

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

МЕЛЬНИЧУК Дмитро Сергійович

**Програмний модуль адаптації html-сторінок для роботи на
мобільних пристроях на основі технології AMP/
Software module for adapting html-pages for mobile devices
based work on AMP technology**

спеціальність: 123 – Комп'ютерна інженерія
освітньо-професійна програма – Комп'ютерна інженерія

Кваліфікаційна робота

Виконав: студент групи КІ-42
Д. С. Мельничук

Науковий керівник
к.т.н Батько Ю.М.

Кваліфікаційну роботу
Допущено до захисту
«__» _____ 20 __ р.

Завідувач кафедри
_____ О.М. Березький

ТЕРНОПІЛЬ - 2021

РЕЗЮМЕ

Кваліфікаційна робота на тему «Програмний модуль адаптації html-сторінок для роботи на мобільних пристроях на основі технології AMP» зі спеціальності 123 «Комп'ютерна інженерія» освітнього ступеня «бакалавр» містить 88 сторінки пояснюючої записки, 12 рисунків, 6 таблиць, 5 додатків. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою кваліфікаційної роботи є розробка програмного модуля адаптації html-сторінок для роботи на мобільних пристроях на основі технології AMP для платформи Magento 2.

Методи дослідження включають методи логічного аналізу принципів роботи платформи Magento 2 та потенційні варіанти для введення інтеграції з технологією AMP.

У проекті надані необхідні розрахунки й матеріали з кодом необхідні для побудови програмного модуля адаптації html-сторінок для роботи на мобільних пристроях на основі технології AMP для платформи Magento 2.

Модуль оптимізації перехоплює згенерований контент сторінок продуктів та CMS сторінок та автоматично обробляє отриманий контент згідно до вимог технології AMP.

Розроблено програмний модуль адаптації html-сторінок для роботи на мобільних пристроях на основі технології AMP

Ключові слова: MAGENTO, AMP, ACCELERATED MOBILE PAGES, OPTIMIZATION, MAGENTO MODULE.

RESUME

Qualification work on the topic "Software module for adapting html-pages for mobile devices based work on AMP technology" in the specialty 123 "Computer Engineering" with a bachelor's degree contains 88 pages of explanatory note, 12 figures, 6 tables, 5 appendices. Volume of graphic material 2 sheets of A3 format.

The purpose of the qualification work is to develop a software module for adapting html-pages for work on mobile devices based on AMP technology for the Magento 2 platform.

Research methods include methods of logical analysis of the principles of the Magento 2 platform and potential options for the introduction of integration with AMP technology.

The project provides the necessary calculations and materials with the code needed to build a software module for adapting html-pages to work on mobile devices based on AMP technology for the Magento 2 platform.

The optimization module intercepts the generated content of product pages and CMS pages and automatically processes the received content in accordance with the requirements of AMP technology.

The software module of adaptation of html-pages for work on mobile devices on the basis of AMP technology is developed

Keywords: MAGENTO, AMP, ACCELERATED MOBILE PAGES, OPTIMIZATION, MAGENTO MODULE.

ЗМІСТ

| | |
|---|----|
| Вступ..... | 9 |
| 1 Технології та програмні засоби створення цифрового контенту для глобальної мережі internet | 11 |
| 1.1 Поняття глобальної мережі інтернет та принципи її роботи | 11 |
| 1.2 Поняття веб-сайту та його різновиди..... | 13 |
| 1.3 Загальний огляд систем управління контентом сайту | 17 |
| 1.4 Висновки до розділу..... | 22 |
| 2 Методи та алгоритми для створення та оптимізації для www..... | 23 |
| 2.1 Система управління сайтами Magento | 23 |
| 2.2 Алгоритми оптимізації роботи сайтів | 26 |
| 2.3 Технологія AMP як спосіб оптимізації | 33 |
| 3 Програмна реалізація модуля адаптації html-сторінок для роботи на мобільних пристроях на основі технології AMP..... | 38 |
| 3.1 Структура програмного модуля AMP | 38 |
| 3.2 Програмні складові програмного модуля AMP..... | 42 |
| 3.3 Тестування та апробація розробленого програмного модуля..... | 56 |
| 4 Техніко-економічний розділ | 59 |
| 4.1 Розрахунок витрат на розробку програмного забезпечення..... | 59 |
| 4.2 Розрахунок експлуатаційних витрат і ціни споживання..... | 64 |
| 4.3 Визначення показників економічної ефективності | 66 |
| Висновки..... | 68 |
| Список використаних джерел..... | 69 |
| Додаток А Лістинг “Magento\Framework\Controller\ResultInterface” | 73 |
| Додаток Б Лістинг класу “ResultInterfacePlugin”..... | 74 |
| Додаток В Лістинг класу “Dathard\Amp\Block\Page\Html\Topmenu” | 79 |
| Додаток Г Довідка про використання | 87 |
| Додаток Е Світлокопія виданої публікації | 88 |

| | | | | | | | | |
|-----------|----------------|----------|--------|------|--|---------------------|------|---------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | | | |
| Змн. | Лист | № докум. | Підпис | Дата | | | | |
| Розробив | Мельничук Д.С | | | | ПРОГРАМНИЙ МОДУЛЬ АДАПТАЦІЇ HTML-СТОРИНОК ДЛЯ РОБОТИ НА МОБІЛЬНИХ ПРИСТРОЯХ НА ОСНОВІ ТЕХНОЛОГІЇ AMP | Літ. | Арк. | Акрушів |
| Перевір. | Батько Ю.М. | | | | | 8 | 91 | |
| Консульт. | Савка Н.Я.. | | | | | ЗУНУ.ФКІТ. КІ-42 | | |
| Н. Контр. | Мельник Г.М. | | | | | | | |
| Затвердив | Березький О.М. | | | | | | | |

ВСТУП

На сьогоднішній день швидкість завантаження web-сайту є досить гострим питанням. Даний параметер є одним з ключових факторів від яких залежить успішність сайту, його позиції в пошуковій видачі, конверсія та інші важливі для бізнесу аспекти.

Така вагомість швидкості завантаження зумовлений тим, що при ідентичних умовах відвідувачі зазвичай обирають той web-сайт, який швидше працює і не змушує витратити зайвий час на очікування. Особливо гостро проблем стоїть для мобільних пристроїв.

Власне на сьогоднішній день є досить велика кількість засобів які допомагають покращити показник швидкості завантаження web-сайтів. Проте більшість з наявних рішень є досить складними і дорогими в процесі їх введення і підтримки, і крім того іноді не дають очікуваних результатів. Крім того є досить велика кількість різноманітних факторів які чинять вплив на швидкість web-сайту і для їх усунення реалізовано каардинально різні засоби.

Серед найважливіших факторів які впливають на швидкість web-сайту в першу чергу можна віднести сам контент сайту. Справа в тому що сучасні сайти мають досить багато громісткого коду для виконання різноманітних анімацій переходів, реалізації меню, стайдерів та багато іншого функціоналу. Даний функціонал робить сайт значно приємнішим в планізовнішнього вигляду, проте нажаль дані скрипти досить погано впливають на швидкість завантаження стоірнок через свою вагу. Крім JavaScript файлів на швидкість також валиває велика кількість CSS коду оскільки зазвичай на сайтах підключається велика кількість зайвого коду який не використовується на даних сторінках. Власне дані фактори є досить вагомими, ароне найбільшим фактором є вага графічного контенту який використовується на сторінках сайту. На сьогоднішній день практично на усіх web-сайтах використовується дуже велика кількість різного роду зображень які можуть займати велику

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 9 |

кількість пам'яті. Власне досить рідко власники переймаються їх оптимізацією, проте не варто зловживати даним фактором оскільки при вірному підході можливо отримати відчутне покращення в швидкості роботи web-сайту.

На сьогоднішній день існує досить великі кількість рішень для оптимізації сатів, детальніше про них йдеться в самій кваліфікаційній роботі, проте досить велика кількість з наявних рішень є вузько спеціалізованими. Через дану специфіку наявнихх рішень іноді з їх впровадженням виникає досить велика кількість проблем які досить дорого обходяться власникам web-сайтів. З метою спрощення даного процесу було розроблено технологію AMP.

Технологія AMP являє собою свого роду чудово збалансовану суміш наявних засобів оптимізації які досить легко використовувати. Власне в даній кваліфікаційній роботі розглядається реалізований спосіб впровадження технології AMP в платформі Magento.

Власне для виконання мети потрібно вирішити наступні задачі:

- проаналізувати принципи роботи технології AMP;
- проаналізувати принципи роботи платформи Magento 2 та підходи реалізації модулів для даної платформи;
- проаналізувати можливі варіанти для введення технології AMP в Magento 2.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 10 |

1 ТЕХНОЛОГІЇ ТА ПРОГРАМНІ ЗАСОБИ СТВОРЕННЯ ЦИФРОВОГО КОНТЕНТУ ДЛЯ ГЛОБАЛЬНОЇ МЕРЕЖІ INTERNET

1.1 Поняття глобальної мережі інтернет та принципи її роботи

Перш ніж розглядати саме поняття інтернету буде доцільно визначитись з поняттям мережі оскільки вона є важливою його складовою. Якщо в загальному то мережею прийнято називати групу підключених комп'ютерів які можуть між собою здійснювати обмін даними. Комп'ютерна мережа має дуже багато спільного з групою людей які працюють в команді, усі вони знають один одного, здійснюють обмін інформацією та спільно координують свою роботу для досягнення певних цілей. Приклад поширеної комп'ютерної мережі зображено на рисунку 1.1.



Рисунок 1.1 – Приклад локальної комп'ютерної мережі

Інтернет є більш глобальним поняттям, він являє собою велику систему з комп'ютерних мереж - мережа мереж, в якій будь-який комп'ютер може, отримати інформацію з іншого комп'ютера, проте за наявності достатнього рівня доступу. Мережу Інтернет можна порівняти з поштовою системою. Обидві дані системи мають подібну структуру і працюють по схожому

принципу. По суті Інтернет як і поштова служба дозволяє людям здійснювати обмін листами, у випадку Інтернету даними листами являються невеликі пакети з даними. Також слід зауважити що доставка поштовою службою може займати досить багато часу, але в випадку Інтернету даний процес триває менше секунди.

В мережі Інтернет працює дуже велика кількість комп'ютерів які використовують різне обладнання та різне програмне забезпечення і це є досить серйозною перешкодою для їх коректного спілкування. Це вимагає використання комунікаційних методів, зрозумілих для всіх підключених комп'ютерів, подібно до того, як людям, які вирости в різних частинах світу, може знадобитися говорити спільною мовою, щоб зрозуміти один одного.

Дану проблему можна вирішити за допомогою стандартизованих протоколів. Якщо коротко протокол - це стандартизований спосіб виконання певних операцій та форматування даних, щоб усі пристрої в мережі могли спілкуватися і розуміти один одного.

Існують протоколи для відправки пакетів між пристроями в одній мережі (Ethernet), для передачі пакетів з мережі в мережу (IP), для забезпечення успішної доставки даних пакетів у встановленому порядку (TCP) та форматування даних для веб-сайтів та програм (HTTP). Оскільки усі підключені до Інтернету ПК та інші пристрої здатні інтерпретувати та розуміти дані протоколи, Інтернет працює незалежно від того хто і що до нього підключається.

Як і згадувалось вище, Інтернет складається з дуже великої кількості апаратних засобів, які забезпечують його стабільну роботу. Деякі з найбільш важливих типів апаратних засобів включають наступне:

- Маршрутизатори відповідають за пересилку пакетів даних в різні комп'ютерні мережі залежно від місця їх призначення.
- Комутатори відповідають за підключення пристроїв, які об'єднані в одну спільну мережу. Вони використовують комутацію пакетів для переадресації пакетів на потрібні пристрої. Також комутатори отримують вихідні пакети від цих пристроїв і передають їх у потрібний пункт призначення.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 12 |

– Веб-сервери - це потужні спеціалізовані комп'ютери, які зберігають і обслуговують вміст сайтів (веб-сторінки, зображення, відео) для користувачів. Слід зауважити що крім місця зберігання інформації сервери також відповідають за DNS запити і виконують інші важливі завдання, щоб підтримувати роботу Інтернету.

1.2 Поняття веб-сайту та його різновиди

Веб-сайти, без сумніву, є найважливішим елементом Інтернету, проте іноді люди не цілком розуміють що це через помилкові визначення даного терміну. Насправді веб-сайт - це сукупність загальнодоступних, взаємопов'язаних веб-сторінок, які мають єдине доменне ім'я. Веб-сайти можуть створюватися та підтримуватися окремою особою, групою, бізнесом чи організацією для різноманітних цілей.

Іноді веб-сайти також називають "веб-сторінкою", дане визначення є помилковим, оскільки веб-сайт складається не з одної, а з декількох веб-сторінок.

На сьогоднішній день в Інтернеті розміщено мільярди веб-сайтів, які конкурують за певну частку уваги, яку люди приділяють своєму веб-перегляду щодня. Проте в загальному всі веб-сайти можна поділити на певні категорії.

Найпопулярніші види веб-сайтів які можна зустріти в інтернеті сьогодні:

- веб-сайти електронної комерції (eCommerce);
- бізнес-сайти;
- розважальні веб-сайти;
- веб-сайти портфоліо;
- медіа-сайти;
- веб-сайт брошури;
- некомерційні веб-сайти;

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 13 |

- навчальні веб-сайти;
- персональні веб-сайти;
- веб-портали;
- вікі або веб-сайти форумів спільноти.

Веб-сайти електронної комерції - це веб-сайти, на яких люди можуть купувати товари. Будь-який веб-сайт, що включає кошик для покупок та спосіб надання інформації про кредитну картку для здійснення покупки, підпадає під цю категорію.

Бізнес-веб-сайт - це будь-який веб-сайт, присвячений представництву конкретного бізнесу. Даний сайт має містити фірмову символіку бізнесу (той самий логотип та позиціонування) і та повідомляти про типи продуктів або послуг, які пропонує бізнес.

На сьогоднішній день у кожної компанії повинен бути власний веб-сайт, це широко розповсюджене очікування. Кожен потенційний клієнт, з яким стикається компанія, просто припустить, що якщо він пошукає більше інформації про компанію у гуглі, він знайде веб-сайт. А якщо вони цього не роблять, це робить бізнес менш професійним.

Веб-сайти електронної комерції - це ділові веб-сайти, але також можна мати діловий веб-сайт, який нічого не продає напяму, а навпаки, заохочує відвідувачів зв'язатися з ним для отримання додаткової інформації або зайти у магазину, якщо вони зацікавлені стати клієнтами.

Зазвичай люди проводять багато часу в інтернеті на різноманітних веб-сайтах. Якщо запитати людину вона з легкістю може навести кілька веб-сайтів, які вона відвідує виключно з метою розваги. Це можуть бути веб-сайти з гумором, такі як The Onion, веб-комікси, або просто веб-сайти із цікавим або цікавим вмістом, такі як BuzzFeed.

Більшість із цих веб-сайтів мають на меті заробляти гроші, як це роблять веб-сайти для бізнесу та електронної комерції, але, як правило, за допомогою реклами, яка відображається на сторінці, а не за рахунок продажу конкретних продуктів або послуг.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 14 |

Веб-сайти портфоліо - це сайти, присвячені демонстрації прикладів минулих робіт. Постачальники послуг, які хочуть показати потенційним клієнтам якість роботи, яку вони надають, можуть використовувати веб-сайт портфоліо, щоб зібрати найкращі зразки виконаної роботи, яку вони зробили. Цей тип веб-сайтів простіший у створенні, ніж діловий веб-сайт, і більше орієнтований на певне завдання - збір зразків робіт.

Цей тип веб-сайтів є найпоширенішим для творчих професіоналів та фрілансерів, які наймаються на основі продемонстрованих навичок і можуть бути більш ефективною альтернативою діловому веб-сайту, який обслуговує подібну увагу.

Якщо говорити про медіа-сайти, то вони в собі містять новини та інші репортажі. Тут є деяке накладання на веб-сайти з розвагами, але веб-сайти з медіа, швидше за все, включатимуть опубліковані фрагменти на додаток до чи замість вмісту, призначеного виключно для розваги.

Зазвичай медіа-сайти заробляють гроші на рекламі, яка відображається на сайті, моделях підписки або за допомогою певної комбінації з них.

Значна кількість медіа веб-сайтів є Інтернет-галуззю засобів масової інформації, яка часто існує в інших формах, таких як телевізійні канали, друковані журнали та газети, але деякі з них є лише в Інтернеті.

Веб-сайти з брошурами - це спрощена форма ділових веб-сайтів. Для підприємств, які знають, що їм потрібна присутність в Інтернеті, але не хочуть вкладати в це багато коштів. Сайти даного типу були найбільш поширені в перші дні інтернету, коли підприємства знали, що їм потрібен веб-сайт, але також очікували, що від нього не залежить їх успіх. На сьогоднішній день люди при підборі товарів та послуг дуже часто користуються інтернетом, саме тому більшість підприємств усвідомлює, що їм потрібно щось більш конкурентоспроможне ніж веб-сайти з брошурами.

Так само, як підприємствам потрібні веб-сайти, щоб представляти їх в Інтернеті, це роблять і некомерційні організації. Некомерційний веб - сайт є найпростішим способом для багатьох потенційних донорів зробити пожертвування і буде першим місцем, куди багато людей звертаються, щоб

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 15 |

дізнатися більше про некомерційну організацію і визначити, чи вони хочуть вони її підтримати.

Веб-сайти навчальних закладів та тих, хто пропонує онлайн-курси, потрапляють до категорії освітніх веб-сайтів. Основна мета даних веб-сайтів - або надання навчальних матеріалів відвідувачам, або надання їм інформації про навчальний заклад. Деякі освітні веб-сайти мають рекламу, на кшталт розважальних та медіа-сайтів. Деякі пропонують підписку або освітні товари для придбання.

Не всі веб-сайти в інтернеті існують для того, щоб заробляти гроші тим чи іншим чином. Багато людей знаходять цінність у створенні персональних веб-сайтів, щоб викласти у світ власні думки. Ця категорія включає особисті блоги, влоги, якими люди ділиться з усім світом. Іноді дані веб-сайти переростають у щось, що заробляє гроші, якщо вони стають достатньо популярними, а той, хто їх розпочав, хоче зробити цей перехід, але вони в основному існують як спосіб поділитися своїми почуттями, думками та мистецтвом з будь-якими друзями та незнайомцями, які можуть цікавитися. Створення веб сайтів даного типу простіше, ніж інших веб-сайтів з даного списку оскільки вимоги від таких сайтів значно нижчі. При створенні даних сайтів люди не турбуються над стимулюванням продаж, або розміщенням реклами, вони просто роблять що їм подобається.

Веб-портали - це часто веб-сайти, розроблені для внутрішніх цілей у бізнесі, організації чи установі. Вони збирають інформацію в різних форматах з різних джерел в одному місці, щоб зробити її доступною для людей, яким потрібно її бачити. Вони часто передбачають персоналізовані перегляди для різних користувачів, які гарантують, що доступна інформація є найбільш корисною для їхніх конкретних потреб. Веб-портали, зазвичай, передбачають більш складне програмування та дизайн, ніж більшість інших веб-сайтів.

Також на сьогоднішній день більшість людей знайомі з вікі через найвідоміший з його представників - Вікіпедію. Проте вікі можна створити на будь-яку. Вікі - це будь-який веб-сайт, на якому різні користувачі можуть співпрацювати над вмістом і разом вносити власні зміни так, як їм зручно.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 16 |

Існують вікі-програми для фан-спільнот, для ділових ресурсів та збору цінних джерел інформації.

Запуск вікі досить простий процес, особливо якщо використовувати існуюче програмне забезпечення або конструктор вікі-сайтів, а не намагатися створити його з нуля.

1.3 Загальний огляд систем управління контентом сайту

На сьогоднішній день значна кількість людей купують речі саме в Інтернеті, щороку кількість таких людей лише зростає. За даними компанії Statista, у 2020 році у всьому світі вже було понад 2 мільярди цифрових покупців, і ця кількість постійно збільшується. Що ще цікавіше, компанія eMarketer прогнозує, що частка електронної комерції в загальному світовому роздрібному продажі досягне 23% до 2023 року (на сьогоднішній день становить біля 15%). З даних показників та прогнозів очевидно що сайти електронної комерції будуть мати значні переваги.

Платформа електронної комерції - це рішення, яке допомагає брендам створювати та керувати своїми інтернет-магазинами. Однак з огляду на велику кількість існуючих та нових, що з'являються щомісяця, стає все складніше вибрати платформу яка дійсно підійде. Лише G2 включає 188 платформ електронної комерції. Саме велика кількість даних платформ ускладнює вибір потрібної.

На рисунку 1.2. зображено статистику по найбільш використовуваних платформах електронної комерції на сьогоднішній день.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 17 |

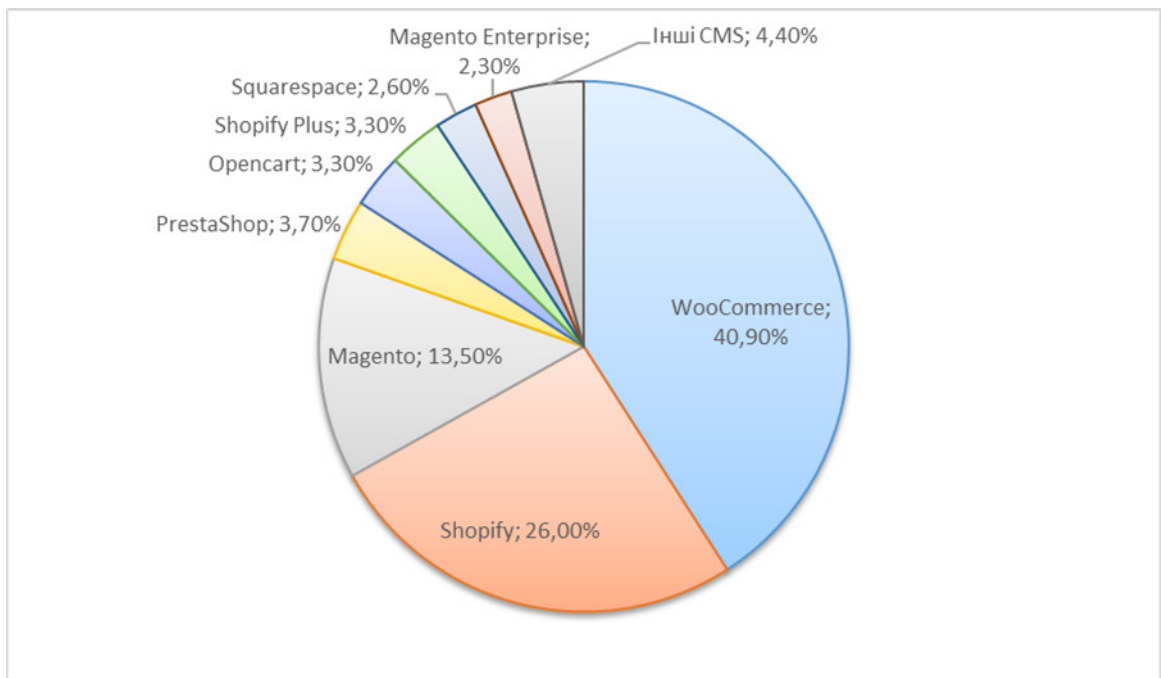


Рисунок 1.2 – Схема популярності платформ електронної комерції

Як видно з статистики на рисунку 1.2. платформа Shopify на сьогоднішній день є одною з найбільш використовуваних для реалізації інтернет-магазинів. Дана платформа здобула таку шалену популярність в першу чергу завдяки простоті керування. Платформа пропонує досить широкий спектр функцій для маркетингу, платежів, продажів, залучення клієнтів та інших видів діяльності.

Ще однією перевагою Shopify є його зручність. Окрім того, що ця система управління вмістом досить проста у використанні, магазини електронної комерції можуть використовувати мобільний додаток Shopify для управління магазином з будь-якого місця.

Також однією з ознак даної системи є її велика гнучкість. Shopify має велику кількість різноманітних цінових планів. Дана особливість дозволяє компаніям підібрати ідеальний для них план.

Проте серед великої кількості переваг даної системи є один вагомий мінус – з власників магазинів стягується плата щоразу, коли вони використовують сторонні платіжні шлюзи. Таким чином, якщо продавець має

намір використовувати сторонні платіжні системи, він повинен бути готовим сплачувати додаткові комісії за транзакції.

Ще одним гігантом на ринку програмного забезпечення для електронної комерції є Magento. Ще у 2017 році понад 250 000 магазинів електронної комерції вже використовували Magento для управління своїм онлайн-бізнесом. Крім того, ця система управління вмістом часто потрапляє до списків найпопулярніших платформ електронної комерції у світі.

Платформа Magento пропонує дуже широкий спектр можливостей по налаштуванню для інтернет-магазинів різного розміру. Magento надає користувачам численні сторонні інтеграції та чудову продуктивність. На додаток до стандартних функціональних можливостей CMS-рішень, також система пропонує багато інструментів для управління каталогами, маркетингом та SEO.

Magento досить універсальна система, проте встановлення та налаштування даного програмного забезпечення електронної комерції є набагато довшим і складнішим порівняно з іншими рішеннями. Проте незважаючи на те, що Magento Open Source можна безкоштовно завантажувати та використовувати, усі інші речі, такі як розробка, теми, розширення та хостинг, можуть вимагати досить багато витрат. Таким чином, Magento більше підходить для відомих магазинів електронної комерції, які мають платоспроможну аудиторію та великий попит.

Говорячи про найкращі та найвідоміші платформи електронної комерції неможливо ігнорувати WooCommerce. WooCommerce - це не повноцінна платформа, а плагін електронної комерції, який використовується разом із CMS WordPress. Однак це не робить його менш потужним, оскільки в на сьогоднішній день його використовує близько 42% усіх веб-сайтів електронної комерції на ринку. WooCommerce - це рішення з відкритим кодом, ядро якого є безкоштовним, але несе витрати, коли йдеться про придбання індивідуальних тем або розширень. Вбудовані функції включають продаж фізичних, цифрових та афілійованих продуктів, підтримку геолокації, кошик покупок реалізований

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 19 |

на базі Ajax, знижки, калькулятор ціни доставки та податків, SEO-оптимізацію та інтеграцію Google Analytics.

Хоча WooCommerce є безкоштовним для тих, хто має веб-сайт на базі WordPress, іншим компаніям доведеться платити за хостинг. Крім того, WooCommerce може вимагати додаткових витрат на придбання плагінів та тем. Проте деякі компанії, які використовують це програмне забезпечення для електронної комерції, мають досить великі ризики через дані сторонні плагіни оскільки не всі з них підключені безпосередньо до WooCommerce, і цей нюанс можуть використовувати хакери.

Ще досить прогресивною платформою електронної комерції на сьогоднішній день є BigCommerce. На сьогоднішній день платформою BigCommerce на сьогоднішній день користується більш ніж 60 000 магазинів різного типу та розміру. Така популярність платформи зумовлена великою кількістю вбудованого функціоналу та можливостями CMS, також дана платформа досить сприятлива для бізнесів електронної комерції які швидко розвиваються. Також слід зауважити що керування даною платформою досить інтуїтивне що приваблює початківців електронної комерції.

Як і Shopify, BigCommerce пропонує досить великий набір планів і це дозволяє клієнтам підібрати найбільш актуальний для нього план. Завдяки цьому система особливо актуальна для підприємств, які хочуть використовувати систему управління вмістом, економічно вигідну та просту в налаштуванні. З іншого боку, після оплати передплати користувачам може знадобитися перейти на більш дорогі плани, оскільки стандартний план BigCommerce не може запропонувати багатий набір тем і розширень.

Ще однією досить хорошою та популярною платформою електронної комерції є OpenCart. Платформа OpenCart є безкоштовною і має досить велику спільноту учасників, тому продукт постійно розвивається та вдосконалюється. До плюсів даної платформи можна віднести велику кількість шаблонів для швидкого налаштування платформи. Також на дану платформу розроблено велику кількість розширень що дозволяє його гнучко налаштовувати. Також OpenCart має інтеграцію із більш ніж 20 платіжними системами та 8 способами

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 20 |

доставки. Як система управління електронною комерцією без щомісячної плати, OpenCart має велику популярність серед новачків із невеликим бюджетом.

Проте дана платформа також має свої мінуси. Одним з найбільших мінусів є проблеми з безпекою та низька продуктивність платформи, проте її можна покращити за допомогою додаткових плагінів та розширень.

Також однією з досить популярних платформ електронної комерції є Prestashop. Багато підприємств використовують дану платформу оскільки за неї не потрібно платити ліцензійну плату за розробку та управління інтернет-магазинами.

Дане програмне забезпечення є досить гнучким та простим в керування. Практично кожен може безкоштовно завантажити даний продукт і без проблем налаштувати його з допомогою різних розширень. Також дана система підтримує широкий спектр інтеграцій, наприклад, пов'язаних із платежами, таких як PayPal або Google Checkout. В PrestaShop крім безкоштовного варіанту також наявний платний - PrestaShop Ready. Власне в PrestaShop Ready вже зроблено усі налаштування системи і продавцю не прийдеться все налаштовувати з нуля.

Завдяки своїм особливостям PrestaShop може бути корисним для інтернет-магазинів, які не хочуть витратити багато часу на розробку рішення для електронної комерції, і це є його вагомою перевагою серед інших. Проте в PrestaShop також є кілька недоліків. Наприклад, продавці можуть витратити додаткові гроші на придбання модулів та доповнень, оскільки не всі вони входять до стандартного набору. Крім того, раніше PrestaShop мав деякі проблеми з модулями платіжних систем, тому потенційно проблеми даного типу можуть повторюватись. Окрім того платформа дозволяє використовувати ряд інших переваг, що значно спрощують роботу як розробника так і користувачів при її використанні. Серед основних її переваг використання сучасних інформаційних модулів, що значно прискорює та здешевлює процес оптимізації та налаштування майбутніх програмних розробок.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 21 |

1.4 Висновки до розділу

У теперішній час Інтернет використовується не лише в якості джерела інформації а та способом спілкування, а і для бізнесу. Власний сайт дозволяє значно допомогти в розвитку підприємства та зробити його більш прибутковим. Власний інтернет сайт дозволяє отримувати більше клієнтів, оскільки на сьогоднішній день велика частина людей шукає інформацію саме в інтернеті і сайт з детальною інформацією про послуги які надаються підприємством.

Крім сайтів візиток сьогодні велику популярність набувають інтернет магазини. Підвищення популярності даного типу сайтів зумовлено тим, що велика кількість людей почали робити покупки саме через інтернет оскільки це зручно, тому інтернет комерція стала досить привабливою нішею для відкриття власного бізнесу. Також підвищення кількості інтернет магазинів зумовлено спрощенням їх запуску. На сьогоднішній день вже не потрібно з нуля розробляти сайт, для цього вже розроблено велику кількість платформ на базі яких можливо реалізувати сайт будь якого формату.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 22 |

2 МЕТОДИ ТА АЛГОРИТМИ ДЛЯ СТВОРЕННЯ ТА ОПТИМІЗАЦІЇ ДЛЯ WWW

2.1 Система управління сайтами Magento

Платформа Magento 2 на сьогоднішній день є однією з найбільш популярних CMS для інтернет магазинів у всьому світі з відкритим кодом. Також слід зауважити що популярність платформи з кожним днем все збільшується. На рисунку 2.1. зображено зібрану ресурсом “Built With” статистику з використання даної платформи.

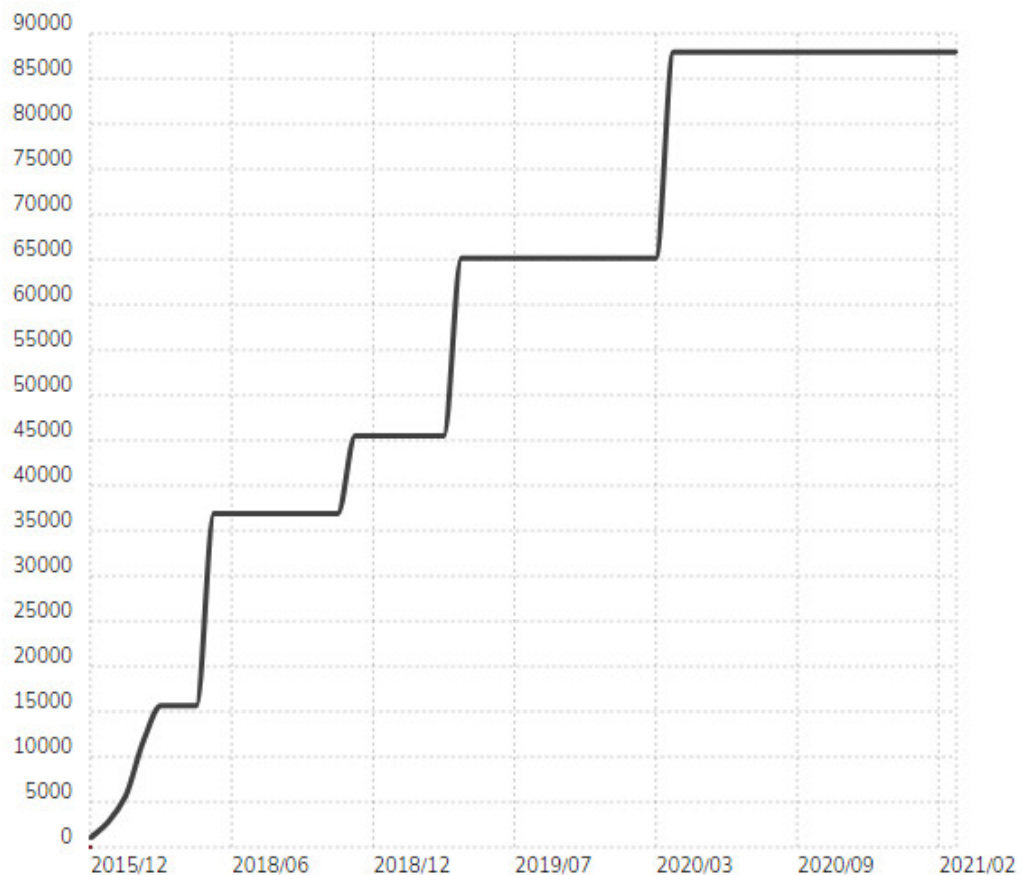


Рисунок 2.1 – Статистика використання платформи Magento 2

Така популярність даної платформи зумовлена завдяки великій кількості функцій які надаються платформою з коробки та можливостями легко та швидко розширювати її функціональність. У середньому підприємства, які

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 23 |

використовують Magento, розширюють свої магазини в 3 рази швидше, в порівнянні з іншими платформами, проте це лише наслідок.

Платформа Magento дозволяє на своїй основі розміщувати на своїй основі безліч веб сайтів та магазинів, і при цьому здійснювати керування ними з допомогою лише однієї адміністративної панелі. Платформа дозволяє незалежно керувати всіма веб сайтами, включаючи їх дизайн, контент, товари, мов, валюти, податкові ставки. Також в Magento дуже розвинені можливості по налаштуванню доступів до окремих розділів адміністративної панелі що дозволяє створювати безліч акаунтів адміністраторів з доступом лише до дозволених їм частин платформи та веб сайтів.

Розробники платформи Magento дуже серйозно відносяться до безпеки і приділяє багато уваги даному параметру. Завдяки вдосконаленому керуванню паролями, запобіганню атак міжсайтового скриптингу і загальному управлінню вразливими даних, Magento швидко реагує і активно розробляє виправлення безпеки для своїх користувачів. Magento Commerce навіть поставляється з Secure Payment Bridge, який забезпечує стандарт безпеки даних PCI, який дозволяє клієнтам безпечно зберігати інформацію про свої кредитні картки для майбутніх транзакцій.

Також платформа Magento з коробки надає велику кількість функцій зручних для SEO, на кшталт оптимізованих URL адрес, Google Sitemap, метадані для продуктів, категорій та звичайних сторінок з контентом, перезапис URL адрес та багато іншого.

Серед основних переваг Magento є велика кількість сторонніх послуг, доступних для впровадження. Платіжні шлюзи, додатки для баз даних, способи доставки та відстеження, спільне використання в соціальних мережах, засоби підвищення продуктивності і багато іншого, що вже розроблено на дану платформу, або ж може бути розроблено згідно до потреб користувачів.

Висока гнучкість Magento в плані розширення функціоналу власне зумовлена самою структурою даної платформи. Платформа працює на базі модульної структури, тобто уся платформа складається з великої кількості модулів, кожен з яких відповідає за певний функціонал (каталоги, продукти,

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 24 |

оплату). Власне завдяки даному підходу будьякий з стандартних модулів Magento може бути без проблем замінений на аналог, або ж взагалі можна доставляти якісь окремі компоненти не вносячи змін в основному кодї платформи.

Сама ж структура фалів Magento зображені на рисунку 2.2.



Рисунок 2.2 – Структура файлів платформи Magento

У директорії “app” розміщуються усі модулі які доставляються на платформу окремо крім стандартних. Файли модулів в даній директорії розміщуються за шляхом в форматі “app/code/<Назва розробника>/<Назва модуля>”. Крім модулів в даній директорії також розміщуються файли тем, файли перекладів та основні файли конфігурацій платформи.

Після інсталяції платформи директорія “app/code” відсутня, оскільки дефольні модулі Magento винесені в окрему директорію “vendor”. Даний підхід дозволяє розділити стандартні модулі платформи від клієнтських модулів.

Також в файловій структурі платформи є ще одна дуже важлива директорія “bin”. В даній директорії зберігається виконуваний скрипт Magento CLI. За допомогою даного скрипта в Magento є можливість запускати команди з допомогою терміналу, які є корисними для очищення кешів, переіндексації, інсталяції модулів та багато чого ще.

В директорії “generated” зберігаються згенеровані файли Magento. Власне в дану директорію поміщаються підготовлені файли мадженти які

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 25 |

використовуються платформою в процесі роботи і це дозволяє пришвидшити роботу.

Директорія “var” використовується для розміщення згенерованих класів, сесій, кешів, резервних копій баз даних та файлів з логами.

Директорія “pub” в собі містить оброблені статичні дані на кшталт мініфікованих js, css файлів та оброблених зображень в різному розмірі.

Завдяки своїй структурі в реалізації платформа Magento є дуже універсальним рішенням для реалізації бізнесу на її основі. Платформа з коробки надає досить багато можливостей, яких цілком вистачає для роботи інтернет магазину, проте якщо даних можливостей буде недостатньо то їх можна легко розширити. Розширення платформи є легким саме завдяки модульній структурі даної платформи, що усуває необхідність безпосередньо вносити зміни в саму платформу (це ще й добре для роботи платформи).

2.2 Алгоритми оптимізації роботи сайтів

Навіть зараз, коли швидкість інтернету у більшості провайдерів перевищує 100 Мбіт/сек, а мобільні оператори активно розвивають 5G, питання про швидкість завантаження сайту залишається дуже актуальним. Даний параметр має безпосередній вплив на конверсію, показник відмов, відвідуваність і інші важливі для бізнесу аспекти. Це зумовлено тим, що при однакових умовах людина обирає той сайт, який швидше завантажується і змушує її чекати.

Досить багато різних компаній проводили дослідження як швидкість сайту впливає на конверсію і всі з них збігаються в думці що оптимальна швидкість завантаження - від 1,5 до 3 секунд. Якщо ж сторінка завантажується довше то більше половини відвідувачів покидають її.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 26 |

Наглядний приклад впливу швидкості завантаження сайту на коефіцієнт конверсії зображено на рисунку 2.3, який базується на дослідженнях проведених ресурсом skilled.co.

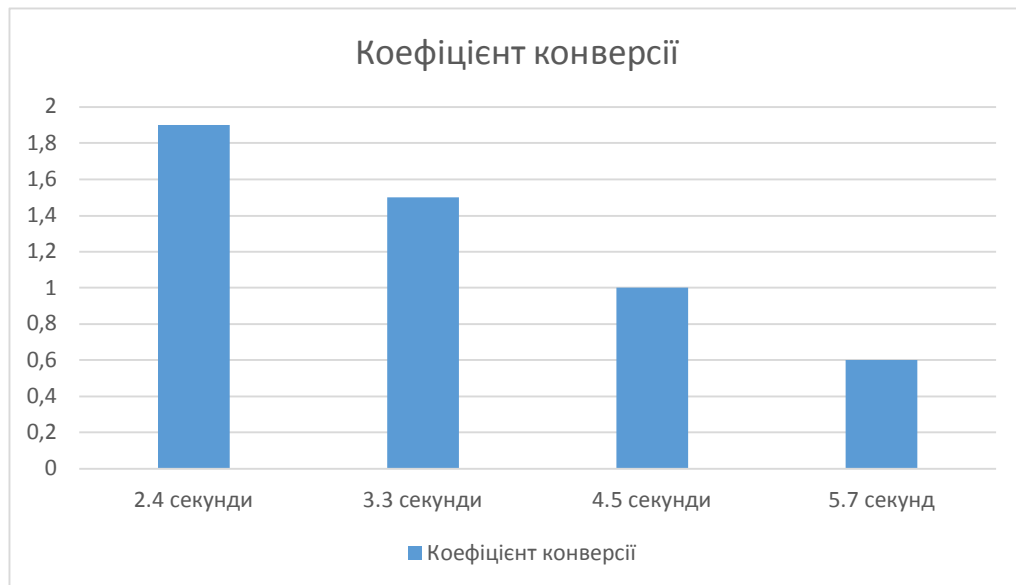


Рисунок 2.3 – Залежність коефіцієнта конверсії від швидкості сайту

Дана ситуація складається через те що відвідувачам простіше повернутися до результатів пошуку і обрати інший сайт ніж чекати поки закінчиться завантаження. При низькій швидкості сайту значно знижується ймовірність що користувачі повернуться на сайт. Компанія Google говорить, що через негативний користувацький досвід ймовірність повторної покупки знижується більш ніж на 60%.

Також слід зауважити що швидкість завантаження сайту також досить сильно впливає на його ранжування. З липня 2018 року Google заявили що швидкість сторінки буде одним з основних факторів ранжування сторінки для мобільного пошуку який займав вагому частину а ранжуванні до теперішнього часу. Недавно в компанії Google заявили що планують внести зміни в систему ранжування сайтів і тепер ранжування буде базуватися на концепції Core Web Vitals.

Core Web Vitals - це набір конкретних факторів, які Google вважає важливими для загальної роботи користувачів веб-сторінки. В ряд даних показників входить найбільша змістовна фарба (LCP), перша затримка введення та сукупний зсув макета

LCP - це час, необхідний для завантаження сторінки з точки зору реального користувача. Іншими словами: це час від натискання на посилання до перегляду більшої частини контенту на екрані.

LCP відрізняється від інших вимірювань швидкості сторінки. Багато інших показників швидкості сторінки (наприклад, TTFB та First Contextual Paint) не обов'язково відображають те, як користувач відкриває сторінку. З іншого боку, LCP фокусується на тому, що насправді має значення, коли мова йде про швидкість сторінки - можливість бачити сторінку та взаємодіяти з нею.

Під параметром “перша затримка введення” мається на увазі час, який потрібен користувачеві для реальної взаємодії з сторінкою сайту. Прикладами взаємодії можуть бути вибір опції меню, клацання на посилання в навігації сайту, введення даних у поле та багато інших операцій.

Показник “сукупний зсув макета” (CLS) - це наскільки стабільною є сторінка при завантаженні (вона ж „візуальна стабільність”). Іншими словами: якщо елементи на сторінці рухаються по мірі завантаження сторінки, то це високий рівень CLS і це дуже погано..

Отже, недоліків в повільних сайтах дуже багато, і при введенні компанією Google індексації сайтів на основі Core Web Vitals такі сайти дуже постраждають в плані видачі гугла. Саме тому оптимізація роботи сайту є дуже важливим питанням, але перш ніж його розглядати слід розглянути чому ж сайти працюють повільно.

Причини повільного завантаження сайтів:

– місце розташування сервера та навантаження на нього. Якщо сервер знаходиться в Австралії, то швидкість доступу до сайту в українця та австралійця буде різною. На користь останнього, звичайно ж;

– не оптимізований код. Веб сторінка складається з html-коду, css-таблиць і JS коду. Якщо JS знаходиться вгорі html-документа – він буде рендеритись

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 28 |

першим, не дозволяючи довантажувати інші елементи сторінки. Захаращеність коду (зайві рядки, не об'єднані таблиці стилів і т. і.) також уповільнює процес обробки коду;

– надлишок плагінів та доповнень. Напевно, сайти великою кількістю примочок у вигляді кнопок соцмереж, показу схожих публікацій, онлайн оплати та всього такого виглядає набагато краще. Хоча й завантажується довше;

– відсутність кешування. У процесі першого відвідування браузер може зберігати такі елементи, як: зображення, шрифти, скрипти й т. п., щоб зменшити кількість звернень до сервера. Без цієї функції сайт постійно звертатиметься до сервера, щоб завантажити їх;

– відео-контент. Велика вага, ненадійний хостинг та відсутність оптимізації робить відео “важким”. При цьому воно може відтворюватися переривчасто або взагалі бути недоступним.

– картинки. Та ж ситуація, що з попереднім пунктом. Невідповідний розмір, кількість та вага негативно впливають на швидкість завантаження інтернет-ресурсу.

Сайти на платформі Magento не позбавлені вищенаведених недоліків, однак розроблено багато методів для усунення даних проблем.

Способи оптимізації швидкості сайту на базі платформи Magento:

- використання лінивого завантаження зображень (Lazy loading);
- використання формату WebP для зображень;
- використання легких тем для сайту;
- вимкнення плоских каталогів (flat catalogs);
- скорочення кількості непотрібних сторонніх розширень;
- використання повносторінкового кешування Varnish;
- перехід до HTTP/2;
- мініфікація контенту.

З усіх типів ресурсів, які використовуються на веб-сайтах найпроблемнішими є зображення, кількість запитів яких займає досить вагому частину з загальної кількості запитів. Власне на рисунку 2.4 зображено

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 29 |

співвідношення різного типу контенту сторінки для середньостатистичного сайту.

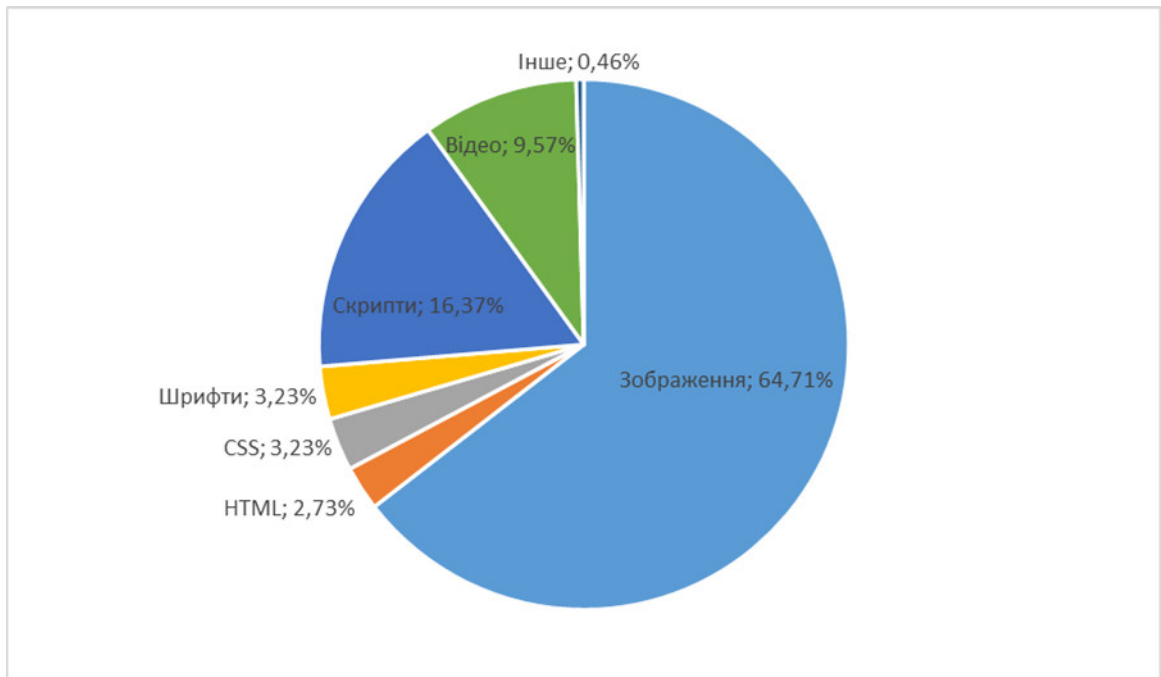


Рисунок 2.4 – Співвідношення контенту на сторінках сайтів

Як видно з діаграми, зображення займають досить вагому частину контенту сторінок, і досить часто саме через них страждає саме швидкість сторінок. З даних причин оптимізація лише самих зображень іноді дає досить вражаючі результати і на сьогоднішній день наявна досить велика кількість підходів для оптимізації цього питання.

Ледаче завантаження - це ефективний спосіб підвищити ефективність роботи сайту, оскільки такий підхід абсолютно позбавляє від необхідності завантажувати всі зображення одночасно при завантаженні сторінки. При ледачому завантаженні зображення завантажуються пізніше, безпосередньо коли в цьому виникає потреба, що фактично зменшує час, необхідний для завантаження сторінки, не шкодячи взаємодії сайту з користувачами.

В даний момент для Magento на жаль немає вбудованого функціоналу який дозволяє реалізувати ледаче завантаження зображень і для його реалізації буде потрібно встановлювати модулі сторонніх розробників.

Крім ледачого завантаження зображень також важливим фактором є сама вага даних зображень. На сьогоднішній день є безліч сервісів для оптимізації різного роду зображень і це досить сильно дозволяє економити ресурси. Проте слід зауважити що різні формати зображень можна оптимізувати лише в певній мірі, притому одні формати піддаються оптимізації добре, а з іншими такого зробити практично неможливо. На сьогоднішній день одним з найбільш оптимізованих форматів зображень для сайтів є формат WebP. WebP як швейцарський армійський ніж форматів оскільки він дуже гнучкий формат зображень, який дуже швидко набирає популярність і, як очікується, в подальшому замінить PNG, JPEG та GIF. В порівнянні з зображеннями у форматі PNG та JPG, зображення WebP, як правило, на 25-35% займають менше місця, при цьому втрата якості зображення майже нульова.

Проте, оскільки це все ще відносно новий формат зображень, WebP стандартний функціонал Magento ще не підтримує даний формат, і для його використання потрібно встановлювати сторонні розширення щоб увімкнути автоматичне перетворення зображень у WebP.

Вагому роль в швидкості сайту відіграє тема яку використовує сайт. Виходячи лише з вражень можна легко сказати коли важка тема значно уповільнює роботу сайту, і користувачі з повільним інтернетом, або застарілим апаратними можливостями від цього лише страждають.

Щоб обійти дану проблему потрібно уважно вибирати теми для Magento, приділяючи більше уваги продуктивності. Однак доволі важко визначити чи дійсно тема легка, тому як правило слід перевіряти, чи демо-версія версія теми здається швидкою, читати більше відгуків користувачів, і щоб бути більш впевненим можна скористатися одним із спеціалізованих сервісів для виміру швидкості завантаження на кшталт PageSpeed Insights.

Раніше найкращим способом для скорочення часу, необхідного для SQL запитів було використання flat каталогів. В мадженті дуже гнучко реалізована структура бази, проте досить часто через це виникали проблеми з швидкістю запитів, особливо в Magento 1. Проблеми з швидкістю виникали за рахунок того що дані розбивались частини і були розміщені в досить велику кількість

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 31 |

таблиць, наприклад лише інформація про продукт може бути розділена більш ніж на 10-15 окремих таблиць. Використання flat каталогів дозволяє об'єднувати частину основних таблиць в одну, це значно пришвидшує запити при роботі з базою даних. Проте починаючи з Magento 2.1 і вище це вже не оптимальна практика і слід утримуватись від будь-якого розширення, яке передбачає використання flat каталогу, оскільки дана функція, спричиняє погіршення продуктивності та інші проблеми з індексацією.

Платформа Magento дуже гнучка за рахунок модульної структури, проте іноді це не дуже добре. Саме завдяки простоті встановлення сторонніх модулів може накопичуватись велика кількість модулів які не використовуються, але залишаються увімкненими в самій мадженті. Крім того, сторонні розширення можуть викликати велику кількість проблем в роботі сайту, якщо їх завантажувати з ненадійних джерел.

Щоб усунути будь-які потенційні проблеми, найкращими методами є зменшення кількості розширень до необхідної та регулярне їх оновлення і завантаження лише з надійних джерел.

Кешування сторінок сайту дозволяє значно зекономити в часі при роботі сайту, це дуже актуально для сайтів, особливо для Magento. В Magento вже є вбудована система кешування яка досить добре справляється з покладеними на неї задачами, проте щоб досягти кращих результатів в швидкості рекомендовано використовувати Varnish кеш. Varnish кеш - це прискорювач веб-додатків з відкритим кодом, який може пришвидшити доставку вмісту в 300-1000 разів. Ось чому самі розробники Magento наполегливо рекомендують Varnish, оскільки це набагато кращий варіант, ніж вбудоване кешування всієї сторінки. Крім того, оскільки Varnish призначений для прискорення HTTP-трафіку, його використання може значно зменшити час відгуку сервера та споживання пропускну здатності мережі на повторні запити. На рисунку 2.5. зображено графік порівняння швидкості завантаження сторінок без використання кеша та з кешами різного типу.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 32 |

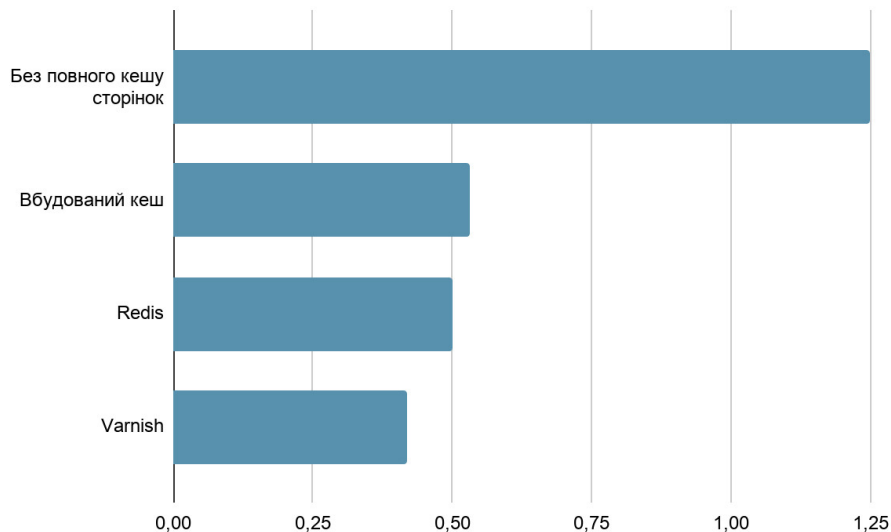


Рисунок 2.5 – Час завантаження сторінок при різних типах кешування

Крім кешування дуже хорошим способом пришвидшення швидкості завантаження сторінок є мініфікації контенту, особливо хороший результат даний метод дає при використанні з кешуванням. Мініфікація, як впливає з назви, є способом зменшення коду сторінок сайту. В процесі мініфікації всі непотрібні частини у файлах HTML, JavaScript або CSS будуть видалені, щоб ресурси сторінки залишались максимально економними. Сьогодні мініфікація є стандартною практикою, і кожен веб-сайт повинен застосовувати її у своїй інфраструктурі, оскільки це, по суті, безкоштовний приріст продуктивності.

Щоб увімкнути мініфікацію для веб-сайту Magento, можна скористатися доступними сторонніми розширеннями, нашталт модуля “Minify HTML CSS JS”, або ж скористатися вбудованими можливостями самої Magento.

2.3 Технологія AMP як спосіб оптимізації

Як і говорилось в попередньому розділі є досить велика кількість факторів які впливають на швидкість сайту. Одним з найбільш вагомих факторів є саме тема яку використовує клієнт та сторонні розширення. В

більшості випадків теми сайтів використовують дуже велику кількість різноманітних JavaScript, CSS та HTML коду який досить погано оптимізований. Особливо гостро дана проблема стоїть для мобільних пристроях, оскільки значна частина контенту на них взагалі не використовується і є зайвою оскільки призначена для ПК. Також в багатьох випадках мініфікація не дає бажаних результатів, оскільки коду всеодно велика кількість. Єдиним варіантом вирішення даної проблеми є видалення зайвих модулів, файлів теми які не використовуються, але завантажуються на сторінку, або ж аналіз та оптимізація даних файлів вручну.

Оптимізація захищеного коду досить складний та довгий процес, який за собою тягне великі витрати. Щоб допомогти позбутися даної проблеми на мобільних пристроях в 2016 році компанія Google запустила проект “Accelerated Mobile Pages” (AMP). Метою даного проекту було пришвидшення продуктивності сайтів на мобільних пристроях, і їм це цілком вдалось. Дана технологія здобула шалену популярність з моменту запуску, і з кожним роком вона стає все популярнішою, про це свідчить графік зображений на рисунку 2.6.

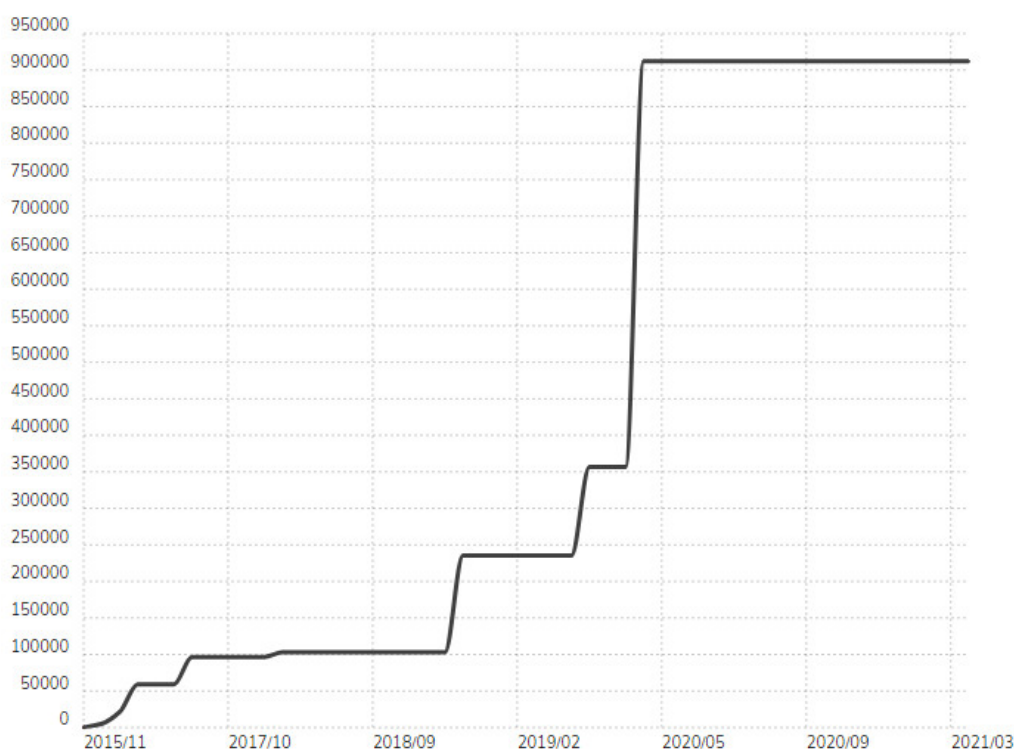


Рисунок 2.6 – Статистика використання технології AMP

Така популярність технології неспроста, AMP призводить до збільшення відвідуваності веб-сайту більш ніж на 10% при збільшенні часу, проведеного на сторінці в 2 рази і це досить вражаючі результати. Веб-сайти електронної комерції, які використовують AMP, отримують збільшення продажів і конверсій більш ніж на 20% порівняно зі сторінками, які не є AMP.

Також слід зауважити що AMP сторінки завантажуються приблизно в чотири рази швидше, ніж стандартні веб-сторінки, плюс користувачі взаємодіють з AMP сторінками на 35% більше, ніж зі стандартними веб-сторінками для мобільних пристроїв:

Саме тому дана технологія здобула вражаючу популярність. На сьогоднішній день AMP забезпечує понад два мільярди сторінок та 900 000 різних доменів.

Більшість людей сьогодні вже зтикалися з сайтами які використовують технологію AMP, але швидше за все не усвідомлюють цього. Єдини що вони могли помітити, це те, наскільки швидко завантажується сторінка після натискання на посилання в пошуку. В більшості випадків на технології AMP можна реалізувати практично повну копію звичайного сайту, але вона буде значно швидшою. Для прикладу на рисунку 2.7 зображено сторінку сайту побудовану на базі AMP та звичайну сторінку сайту.



Рисунок 2.7 – Порівняння AMP сторінки з звичайною

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 35 |

Причиною того, що AMP сторінки завантажуються дуже швидко, полягає в тому, що AMP значно обмежує HTML/CSS та JavaScript, дозволяючи швидше відображати мобільні веб-сторінки.

AMP сторінки створюються на базі наступних трьох компонентів:

- AMP HTML;
- AMP JavaScript;
- кеш-пам'ять AMP.

AMP HTML - це, по суті, HTML, лише з деякими обмеженнями для надійної роботи. Для прикладу найпростіший файл AMP HTML виглядає наступним чином.

Більшість тегів в AMP HTML є звичайними HTML тегами, однак деякі HTML теги замінюються на теги, спеціальні для технології AMP. Дані спеціальні теги називаються AMP-компонентами HTML, і вони полегшують реалізацію загальних шаблонів тегів. AMP сторінки виявляються пошуковими системами та іншими платформами за допомогою тегу HTML.

В технології AMP жорстко обмежена можливість використовувати сторонні JavaScript скрипти, натомість AMP має власну бібліотеку JavaScript скриптів, яких цілком достатньо для побудови компонентів сторінок.

Бібліотека AMP JS забезпечує швидке відображення сторінок AMP HTML. Бібліотека реалізує всі найкращі практики роботи AMP, такі як вбудований CSS та активація шрифтів, керує завантаженням ресурсів та надає власні HTML теги для швидкого виведення сторінки.

AMP JS робить все із зовнішніх ресурсів асинхронним, так що ніщо на сторінці не блокує рендеринг. JS також використовує інші технології продуктивності, такі як пісочниця всіх фреймів, попередній розрахунок макета кожного елемента сторінки до завантаження ресурсів та вимкнення повільних селекторів CSS.

Одним з вагомих плюсів використання технології AMP є Google AMP кеш-пам'ять. Google AMP кеш-пам'ять використовується для обслуговування кешованих сторінок AMP HTML. AMP Cache - це мережа доставки вмісту на основі проксі, яка використовується для доставки всіх дійсних документів

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 36 |

AMP. Кеш отримує сторінки AMP HTML, кешує їх і автоматично покращує продуктивність сторінки.

AMP Cache має вбудовану систему перевірки, яка підтверджує, що сторінка гарантовано працює, і що вона не залежить від зовнішніх факторів, які можуть уповільнити роботу сторінки. Система перевірки працює на ряді тверджень, які підтверджують, що розмітка сторінки відповідає специфікаціям AMP HTML.

Дані три базові компоненти AMP працюють в унісон, що дозволяє швидко завантажувати сторінки.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 37 |

3 ПРОГРАМНА РЕАЛІЗАЦІЯ МОДУЛЯ АДАПТАЦІЇ HTML-СТОРІНОК ДЛЯ РОБОТИ НА МОБІЛЬНИХ ПРИСТРОЯХ НА ОСНОВІ ТЕХНОЛОГІЇ AMP

3.1 Структура програмного модуля AMP

Як відомо платформа Magento дозволяє розміщувати на своїй основі досить громісткі сайти інтернет магазинів. Зазвичай сайти даного типу мають досить велику кількість різноманітних сторінок з яких власне і складається сам сайт. Власне через велике різноманіття даних сторінок іноді виникає багато проблем з редагуванням контенту даних сторінок.

Щоб спростити роботу з даними сторінками на основі їх контенту їх було розділено на групи. Даний розподіл надає досить багато можливостей по роботі з сторінками сайту і спрощує розуміння його загальної структури. Найбільш основними типами сторінок в платформі Magento є:

- сторінки продуктів;
- сторінки каталогу продуктів;
- сторінки пошуку продуктів;
- сторінки оформлення замовлення;
- CMS сторінки.

Власне дані типи сторінок складають основу будь якого сайту на платформі Magento, але крім них є ще досить великі кількості інших типів.

Даний розподіл сторінок по типах дозволив досить гнучко редагувати їх контент під потреби будь якого сайту практично без обмежень. Дана гнучкість реалізована в специфічному підході до побудови сторінок в Magento який базується на спеціальних макетах. Макети в Magento являють собою спеціальні XML файли, в яких сторінки схематично розділено на блоки. Завдяки даному поділу сторінок на блоки можливість вивести будь який контент в будь якому місці сторінки без потреби змінювати щось крім даного файлу. Власне і робота даних макетів базується на поділі сторінок на типи. Для кожного типу сторінок

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 38 |

існує свій макет, даний підхід дозволяє вносити зміни одразу на всі сторінки певного типу.

Підхід з групуванням сторінок і використанням макетів досить гнучкий, але щоб внести якісь зміни обов'язково потрібно змінювати самі файли модулів або теми і це не завжди зручно. Проблема в тому що на сайті є деякі сторінки контент яких досить часто змінюється і в адміністратора має бути можливість легко і швидко це робити. Да таких сторінок відноситься головна сторінка сайту, сторінка контактів, політики конфіденційності та інші. Подібні сторінки було віднесено в до окремого типу під назвою "CMS сторінки" який редагується переважно лише з панелі адміністратора.

Платформа Magento дозволяє досить швидко запустити інтернет магазин за рахунок стандартної теми "Luma". Тема "Luma" в собі містить стилізацію вищенаведених типів сторінок і в результаті повністю робочий сайт з оформленими сторінками. Також слід зауважити що дана тема не дуже часто використовується в оригінальному виді на сайтах, а частково редагується під потреби самих сайтів та часто слугує базою при розробці інших тем.

Практично завжди при розробці сторонніх модулів розробники використовують стандартну тему "Luma", проте для розробки її мало. Платформою Magento передбачено велике різноманіття продуктів різного типу, які не входять в саму тему оскільки вона містить лише зовнішнє оформлення сайту. Проте при розробці дуже важливо враховувати велику кількість різноманітних факторів які виникають з платформою в процесі її експлуатації.

З даної причини розробниками платформи було випущено спеціалізований модуль "Magento_SampleData". Даний модуль містить в собі зразок реального магазину з більш ніж 250 продуктами, більша частина з яких є налаштовуваними. Magento надає досить велику кількість різноманітних типів продуктів:

- звичайні продукти (Simple Product);
- налаштовувані продукти (Configurable Product);
- згруповані товари (Grouped Product);
- віртуальні і завантажуванні товари;

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 39 |

– збірні продукти (Bundle Product).

Звичайні продукти являють собою фізичний елемент з одним SKU (унікальним артикулом), це найпростіший в розумінні тип продукту з наведених вище. В свою чергу налаштовуваний продукт являє собою єдину картку товару зі списками опцій для кожного варіанту. Однак, кожен варіант являє собою окремий простий продукт з окремим SKU, який дозволяє відстежувати кількість доступних одиниць даного товару. Звичайний вигляд продуктів даного типу зображено на рисунку 3.1.

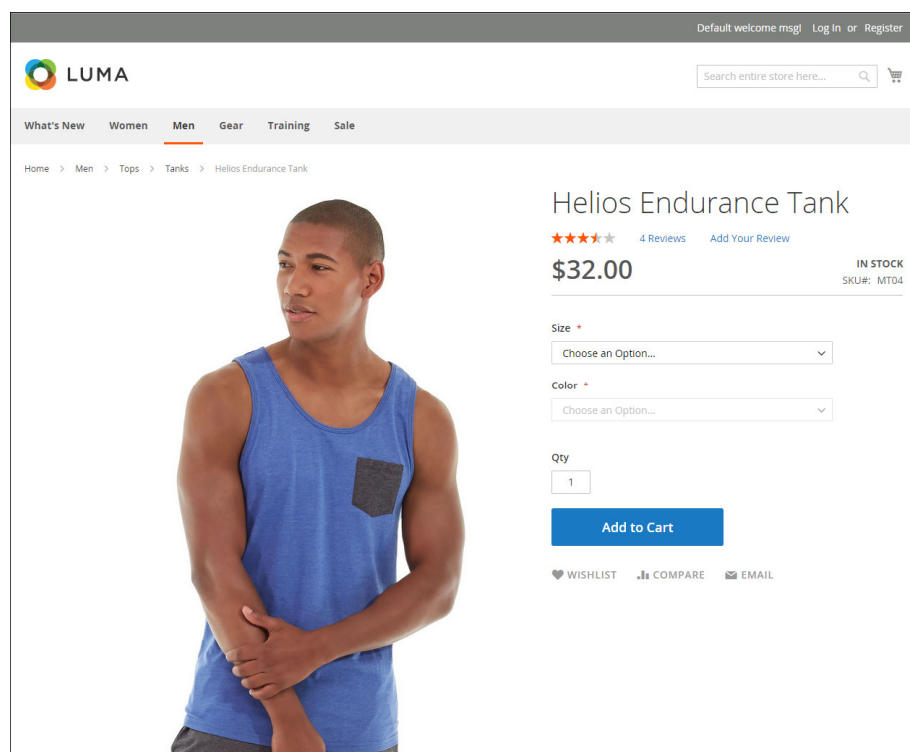


Рисунок 3.1 – Приклад збірного продукту

Згруповані продукти ж являють собою кілька окремих товарів у вигляді групи. Дані продукти можливо придбати окремо, або ж у вигляді групи. Згруповані товари дозволяють об'єднати в 1 картку кілька абсолютно різних товарів. У товарів можуть бути різні набори атрибутів або категорії, також будь-який товар можливо купити окремо. Продукт даного типу зображено на рисунку 3.2.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 40 |

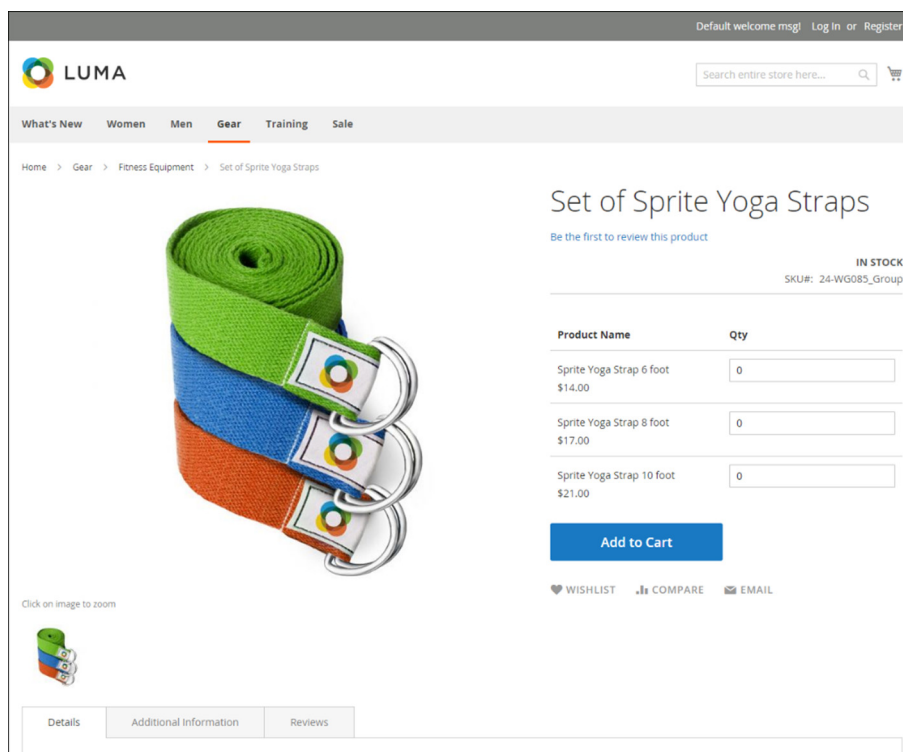


Рисунок 3.2 – Приклад згрупованого продукту

Також Magento містить два досить цікаві і схожі типи продуктів - віртуальні і завантажувальні товари. Дані типи продуктів досить схожі і іноді важко зрозуміти в чому різниця між ними. Віртуальні і завантажувальні товари використовуються для продажу не фізичних товарів, які можна придбати тільки зареєструвавшись на сайті. Різниця в тому, що купуючи завантажувальні товари, ви отримуєте посилання на скачування. Віртуальні товари не дають ніяких посилань, просто потрібна інформація про покупку фіксується за конкретним покупцем. Віртуальні товари використовуються для продажу ліцензій, членства в клубі, підписок на журнали та іншого. Завантажувальні товари підходять для продажу софта, аудіо, відео матеріалів та всього, що потрібно завантажувати після покупки.

Як видно Magento надає досить велику кількість різноманітних типів продуктів які мають свою специфіку використання та роботи з ними. З даної причини при розробці модулів для платформи Magento слід ретельно перевіряти чи не здійснено модулем якогось негативного впливу на роботу даного функціоналу оскільки він життєво необхідний для інтернет магазинів.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 41 |

Власне тому в процесі розробки активно використовується зразки даних модуля “Magento_SampleData”.

Також крім зразків даних про продукти модуль містить різноманітні категорії продуктів, правила рекламних цін, CMS сторінки, різноманітні банери та інше. Також слід зауважити що дані зразки даних використовують тему “Luma”.

Саме з наведених вище причин при розробці дипломного проекту було використано тестове наповнення даними для платформи Magento оскільки воно дозволяє повністю відтворити реальний сайт з його проблемами і відповідно має досить актуальні показники.

3.2 Програмні складові програмного модуля AMP

Як відомо, технологія AMP має досить жорсткі правила щодо побудови HTML сторінок. В даній технології сильно урізано кількість HTML які можна використовувати, і крім того технологією передбачені власні спеціалізовані HTML теги на кшталт “amp-img” та інших. Також слід зауважити що технологія AMP має обмеження щодо CSS коду який використовується на сторінці. Якщо бути точнішим то весь CSS код має бути розміщений безпосередньо в коді самої сторінки, а не в окремому файлі, також його вага не має перевищувати 50KB. Також в технології AMP Заборонено використання стороннього JavaScript коду.

Усі наведені вище обмеження значно ускладнюють впровадження технології AMP на сайтах які працюють вже досить тривалий час. Проблема в тому що такі сайти на платформі Magento вже мають досить велику кількість сторонніх модулів та власні теми які швидше за все змінено під потреби сайту. Саме з даних причин зазвичай на подібних сайтах нагромаджена велика кількість HTML, JavaScript та CSS коду який досить погано оптимізований і займає багато місця. Саме з даних причин вже поточний код досить важко

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 42 |

переробити щоб він був більш оптимізований та підпадав під вимоги технології AMP.

Власне є два способи виходу з даного положення для впровадження технології AMP на сайт на платформі Magento:

- розробка для сайту нової теми яка буде базуватися на технології AMP;
- оптимізація вже готових сторінок відповідно до вимог технології AMP.

Обидва дані варіанти досить дієві, проте варіант з створенням нової теми хоч і найменш проблематичний, але він досить затратний і не дуже універсальний для використання на різних сайтах. Більш універсальний варіант це оптимізація вже готових сторінок. Такий спосіб оптимізації можна робити як вручну, переглядаючи весь код сайту і оптимізуючи кожен елемент окремо, так і автоматично при генерації сторінок. Варіант із ручним способом досить не поганий, але він є досить затратним і не рентабельним якщо говорити про його впровадження на різних сайтах, оскільки все прийдеться робити заново.

Найбільш зручним, практичним і універсальним є спосіб з автоматичною оптимізацією поточних сторінок при їх генерації. Даний спосіб дасть змогу берегти основну структуру сторінки та в загальному її вигляд, також буде досить просто додатково внести правки відповідно до потреб сайту. Даний спосіб оптимізації, якщо говорити про платформу Magento, на базі спеціалізованого модуля. Такий спосіб також досить універсальний оскільки модуль можливо легко встановити на будь який сайт який працює на даній платформі без якихось додаткових змін в коді.

Створити найпростіший модуль для платформи Magento досить просто, для цього він має містити два основні файли:

- `module.xml`;
- `registration.php`.

Файл “`module.xml`” міститься в директорії “`etc`” в кореневій директорії модуля. Даний файл є обов’язковим і містить інформацію про назву модуля, його версію та залежності. Файл “`module.xml`” в розробленому мною модулі має наступний вигляд:

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 43 |

```
<?xml version="1.0" ?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:framework:Module/etc/module.xsd">
    <module name="Dathard_Amp" setup_version="1.0.0"/>
</config>
```

Також одним з обов'язкових файлів для роботи модуля є файл "registration.php". Даний файл є важливим оскільки саме він повідомляє Magento як знайти модуль при компіляції файлів. Файл "registration.php" має стандартизований формат, який дотримується однакової схеми для всіх модулів. Єдине, що змінюється – це назва модуля, в моєму випадку назвою є "Dathard_Amp". Файл "registration.php" в розробленому мною модулі має наступний вигляд:

```
<?php
use Magento\Framework\Component\ComponentRegistrar;
ComponentRegistrar::register(
    ComponentRegistrar::MODULE,
    'Dathard_Amp',
    __DIR__
);
```

Проте для реалізації автоматичної оптимізації сторінок мало створити простий модуль, в даному випадку потрібно реалізувати аналіз згенерованого контенту сторінки та його правки по вимог технології AMP. Для реалізації даного функціоналу необхідно безпосередньо втручатися в саму логіку платформи яка бере участь в генерації сторінок, проте заборонено вносити зміни в файли самої платформи. Для такого випадку в Magento реалізовано кілька методів для внесення подібних змін, це можна зробити використовуючи:

- плагіни;
- спостерігачів;
- переваги (preference).

Плагіни або перехоплювачі - це класи, які змінюють поведінку функцій публічного класу, перехоплюючи виклик функції та виконуючи код до, після або навколо цього виклику функції. Це дозволяє замінювати або розширювати

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 44 |

поведінку оригінальних, публічних методів для будь-якого класу чи інтерфейсу.

Даний підхід перехоплення зменшує кількість конфліктів між розширеннями, які змінюють поведінку одного і того самого класу або методу. Реалізація класу палагіну змінює поведінку оригінальної функції класу, але не змінює сам клас. Magento викликає дані перехоплювачі послідовно відповідно до налаштованого порядку сортування, щоб вони не конфліктували між собою.

Також в Magento є таке поняття як події та спостерігачі. Спостерігачі - це певний тип класів Magento, який може впливати на загальну поведінку, продуктивність або змінювати ділову логіку платформи. Спостерігачі виконуються, коли менеджер подій надсилає подію, яку вони налаштовані спостерігати. В совою чергу події генерується модулями, коли відбуваються певні дії на кшталт запису/читання з бази даних, різноманітних маніпуляцій з даними та власне самим контентом сторінок.

Також в Magento є ще один досить потужний спосіб внесення змін в роботу платформи з допомогою переваг. Завдяки даному способу можливо замість оригінального класу платформи змусити використовувати інший клас в який наслідуються від оригінального. Даний спосіб доситьзручний за у випадках коли потрібно змінити значу частину оригінального коду, або ж коли неможливо внести необхідні правки з допомогою плагінів та спостерігачів.

Власне при реалізації модуля оптимізації я активно використовував лише плагіни та спостірігачів, оскільки використання переваг є крайнім заходом, якого по можливості бажано уникати.

Отже, реалізація автоматизованої оптимізації сторінок сайту вимагає аналіз та правку коду безпосередньо перед його вводом користувачам. В платформі Magento HTML не виводиться безпосередньо в процесі виконання самого запиту, а нагромаджується в спеціальну змінну і виводиться коли процес обробки було повністю закінчено. За вивід нагромадженого контенту відповідає стандартний інтерфейс платформи Magento “Magento\Framework\Controller\ResultInterface”, який представлено в додатку А.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 45 |

Якщо бути точнішим то в даному випадку для реалізації поставленої задачі мене зацікавив метод `renderResult`, оскільки саме він відповідає за роботу з вже згенерованим контентом та його виводом користувачам.

Найзручніше для роботи з вже згенерованим контентом буде працювати вже після того як виконається метод `renderResult`, щоб контент вже був абсолютно згенерований. Для даної задачі мною було вирішено використати плагін. Платформою `Magento` надається три типи плагінів: `before`, `around` та `after`.

Для роботи з результатами виконання функції можливо використати як `after` так і `around` плагіни, проте `around` плагіни слугують свого роду обгорткою для оригінального методу. `Around` плагіни запускаються перед безпосереднім виконанням оригінального методу, виконуючи певну логіку, так і вже після того як виконався оригінальний метод.

В моєму випадку не було потреби виконувати свою логіку перед виконанням методу `renderResult`, потрібно лише виконати операції з результатами його виконання, тому було використано плагін типу `after`.

Для створення плагіну в `Magento` в першу чергу його слід оголосити в спеціальному файлі `di.xml`. Файл `di.xml` є конфігураційним файлом, в якому визначається які залежності вводяться в диспетчер об'єктів `Magento`, на кшталт плагінів, перезапису оригінальних об'єктів та багато чого ще. Код в файлі `di.xml` для реєстрації плагіну на інтерфейс `Magento\Framework\Controller\ResultInterface` має наступний вигляд:

```
<?xml version="1.0" ?>
<config xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="urn:magento:framework:ObjectManager
/etc/config.xsd">
    <type name="Magento\Framework\Controller\ResultInterface">
        <plugin name="controller-result-interface-amp"
type="Dathard\Amp\Plugin\Framework\Controller\ResultInterfacePlugi
n"/>
    </type>
</config>
```

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 46 |

Крім рестрації плагіну в “di.xml” також потрібно вказати назву класу в якому розміщено реалізацію плагіну, такі файли, як правило, розміщені в директорії “Plugin” в кореневій директорії модуля. В даному випадку це клас “Dathard\Amp\Plugin\Framework\Controller\ResultInterfacePlugin” який представлено у додатку Б.

В технології AMP є досить багато обмежень по використанню HTML тегів, тому в згенерованому контенті сторінок їх слід замінити на дозволені аналоги. Проаналізувавши дану задачу та можливі варіанти її вирішення було обрано реалізувати даний процес базуючись на регулярних виразах.

В технології AMP заборонено використання тегів video, audio та ui оскільки технологією передбачено їх більш оптимізовані аналоги amp-video та amp-audio. В розробленій реалізації для заміни даних тегів було вирішено замість регулярних виразів скористатися стандартною функцією php “str_ireplace”, оскільки вона є більш оптимізованою і оптимальною для поточної задачі. Реалізований код має наступний вигляд:

```
$html = str_ireplace(
    ['<video', '/video>', '<audio', '/audio>', '<ui', '/ui>'],
    ['<amp-video', '/amp-video>', '<amp-audio', '/amp-audio>',
    '<ul', '/ul>'],
    $html
);
```

Також в AMP заборонено використання елементів із сторонніх ресурсів, наприклад ютубу та vimeo. Повністю прибрати їх із сторінок сайту не дуже хороша ідея, оскільки це може значною мірою поламати вигляд сайту а AMP версії, оскільки вона не передбачалась при початковій розробці. Проте проаналізувавши можливості що надаються технологією AMP було вирішено для використання відеоконтенту з даних ресурсів скористатися спеціальними тегами “amp-youtube” та “amp-vimeo”. Дану заміну було вирішено виконувати з використанням регулярних виразів. В розроблюемому програмному модулі був реалізований наступний код для виконання який спрацьовує безпосередньо перед виводом контенту:

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 47 |

```

$html = preg_replace(
    '</iframe[>]*?youtube\.com\/embed\/([>|\"|\'|?]+)[>]*?>\s*<\/if
rame\s*>/i',
    '<amp-youtube data-videoid=\"$1\" layout=\"responsive\"
width=\"480\" height=\"270\"></amp-youtube>',
    $html
);
$html = preg_replace(
    '</iframe.+?player\.vimeo\.com\/video\/(.*) (?:\|\"|.)+><\/iframe>/
s',
    '<amp-vimeo data-videoid=\"$1\" layout=\"responsive\" width=\"480\"
height=\"270\"></amp-vimeo>',
    $html
);

```

Крім відео з сторонніх ресурсів AMP забороняє підключення CSS, JS та інших сторонніх файлів, оскільки їх велика кількість значно уповільнює швидкість сайту. Дана проблема досить актуальна для Magento оскільки платформа має модульну структуру, в якій кожен з модулів може додавати власний HTML, CSS та JavaScript код в будь яке місце на сторінках сайту. Власне тому досить часто в Magento виникає нагромадження зайвого і важкого коду.

Для вирішення даної проблеми було реалізовано наступне рішення яке дозволяє видаляти подібний код з контенту сторінок:

```

$html = preg_replace(
    '/\s+(?:style|align|hspace|vspace|itemprop|itemscope|itemtype|data
url|on\w{4,12}|border|vocab|typeof|container|usemap|cellpadding|ce
llspacing|nowrap)\s*=\s*(?:\"[^\"]*\"|\'[\^\\']*\\')/i',
    '',
    $html
);

$html = preg_replace('</link[ ]*rel=\"stylesheet\"[>]+\>/', '',
$html);
$html = preg_replace('#\"(javascript:\s*[^\"]*)\"#isU', '#nohref',
$html);
$html =
preg_replace('#<script((?!ampproject|application\/ld\+json|applica
tion\/json).)*>.*</script>#isU', '', $html);

$html = preg_replace(

```

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 48 |

```

[
'#<script((?!ampproject|application\/ld+json|application\/json).)
*>.*</script>#isU',
    '#<style(?:?!amp-).)*?>[<]*?<\/style\s*?>#isU',
    '#<map.*>.*<\/map>#isU',
    '#<link\s+href="https?:\/\//schema.org\/[a-zA-Z0-9_\-
\/\?&]*"\s?\/?>#isU',

'#(?:<col\s+[>]*(width=(?:"[^"]*"|'[^']*')\s*))#isU'
],
'',
$html
);

```

Ще слід зауважити що технологією AMP передбачено використання спеціальних тегів для виводу зображень на сторінці, стандартні HTML html теги для роботи з зображеннями технологією заборонено використовувати. Для виводу зображень слід використовувати спеціальний тег “amp-img”, з допомогою якого спеціалізований JavaScript код оптимізовує процес завантаження зображень.

Також технологією передбачено спеціальні правила для роботи з розмірами зображень. Якщо бути точнішим то заборонено змінювати розміри зображень безпосередньо з CSS коду, це можливо робити виключно через атрибути “width” та “height”. Дане обмеження дозволяє уникнути зсуву контенту в процесі завантаження сторінки резервуючи необхідне місце заздалегідь. Проблема зсуву контенту також негативно впливає на показники сторінки при скануванні пошуковими ботами і даний підхід допомагає усунути і її.

Проте в Magento не дуже зручно проставляти атрибути з розмірами для кожного зображення, тому було реалізовані стандартні розміри 290 на 100 пікселів, цього зазвичай достатньо для мобільних пристроїв. Також в процесі розробки було реалізовано можливість окремо задавати нестандартні розміри для зображень з допомогою спеціальних атрибутів “data-width-amp” та “data-height-amp”. Значенні даних атрибутів при обробці коду підставляються замість дефолтних з допомогою наступного коду Який динамічно визначає встановленні значення та автоматично підставляє їх у вірні атрибути:

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 49 |

```

    /* Replace width value with data-width-amp value */
    $html =
preg_replace('#(<img\s+[>]*) (?:width=(?:"\w+"|\'\w+\')) ([>]*) (?:
data-width-amp="(\w+)") ([>]*>)#isU', '$1 width="$3" $2 $4',
$html);
    $html =
preg_replace('#(<img\s+[>]*) (?:height=(?:"\w+"|\'\w+\')) ([>]*) (?:
:data-height-amp="(\w+)") ([>]*>)#isU', '$1 height="$3" $2 $4',
$html);

    /* replace data-width-amp with width */
    $html = preg_replace('#(<img\s+[>]*) (?:data-width-
amp="(\w+)") ([>]*\/?>)#isU', '$1 width="$2" $3', $html);
    $html = preg_replace('#(<img\s+[>]*) (?:data-height-
amp="(\w+)") ([>]*\/?>)#isU', '$1 height="$2" $3', $html);

    /* Add height & width if not exists */
    $html = $this->addImageAttribute('height', 100, $html);
    $html = $this->addImageAttribute('width', 290, $html);

```

В ході роботи з атрибутами активно використовується метод “addImageAttribute”. Даний функціонал було винесено в окремий метод оскільки щоб уникнути дублювання схожого по принципу роботи коду. Власне сам міст даного методу має наступний вигляд:

```

private function addImageAttribute(string $attributeName, string
$attributeValue, string $html): string
{
    $regex = "#<img[>](?!.*$attributeName)[>]+>#";
    if (preg_match_all($regex, $html, $imagesWithoutAttr)) {
        foreach ($imagesWithoutAttr[0] as $imageHtml) {
            $imageWithAttr = str_replace(
                '<img',
                "<img $attributeName=\"\$attributeValue\"",
                $imageHtml
            );

            $html = str_replace($imageHtml, $imageWithAttr,
$html);
        }
    }

    return $html;
}

```

Власне ж сама заміна тегів “img” на більш оптимізовані “amp-img” здійснюється наступним чином:

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 50 |

```
$html = preg_replace('#(?:<img\s+)(.*?)(?:\>|>)#is', '<amp-  
img $1></amp-img>', $html);
```

Власне більша частина по оптимізації була виконана з допомогою наведеного вище коду, проте слід зауважити що є досить багато елементів які неможливо оптимізувати подібним чином. До подібних елементів відноситься меню сайту оскільки в більшості випадків для його реалізації на сайтах використовують сторонні модулі яких на сьогоднішній день досить велика кількість, з цієї причини коректно оптимізувати їх код практично неможливо. Саме тому було вирішено реалізувати меню на базі методів які надаються технологією AMP та стандартного функціоналу платформи Magento.

Проте слід зауважити що зазвичай меню сайту на мобільних пристроях міститься в спеціальній панелі яку можна відкрити натисканням спеціальної кнопки. Приклад такого меню зображено на рисунку 3.3.

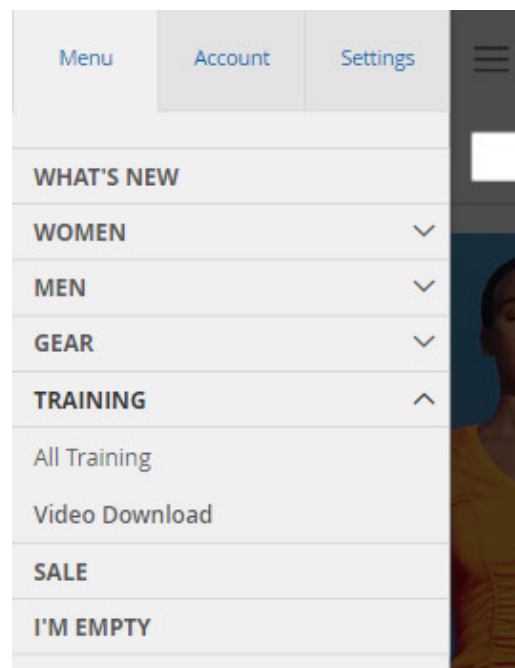


Рисунок 3.3 – Приклад меню сайту на мобільних пристроях

На сьогоднішній день відображення меню сайту в подібному форматі є досить поширеним за рахунок економії місця на сторінці та зручності його використання. Проте зазвичай ний формат виведення меню реалізовується на базі JavaScript коду, досить часто на базі бібліотек на кшталт jQuery. Як

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 51 |

згадувалось раніше, технологією AMP заборонено використання стороннього JavaScript коду, тому використати даний підхід нажаль не вийде. Проте розробниками AMP було передбачено можливість виводити контент подібним чином з допомогою спеціального тегу “amp-sidebar”. Тег “amp-sidebar” приховує метавміст, призначений для тимчасового доступу, його також можна відкривати за закриття натисканням спеціальних кнопок за його межами. З даних причин в реалізації навігації сайту було використано даний тег.

За реалізацію самої навігації сайту, в розробленому модулі оптимізації, відповідає об’єкт “Dathard\Amp\Block\Page\Html\Topmenu” який надано у додатку В.

Реалізований об’єкт “Dathard\Amp\Block\Page\Html\Topmenu” наслідується від стандартного об’єкту Magento “Magento\Theme\Block\Html\Topmenu”, який використовується для побудови меню. На даному етапі було вирішено в якості основи використати стандартну реалізацію даного функціоналу, а не реалізовувати все з нуля щоб в подальшому уникнути якомога більшої кількості конфліктів.

Власне щоб для формування власного меню сайту було вирішено переписати кілька методів оригінального об’єкту, які власне відповідають за побудову і вивід навігації. Якщо бути точнішим то було перероблено методи “_addSubMenu” та “getHtml”. Найважливішим з даних методів є метод «getHtml». Даний метод слугує свого роду вхідною точкою, оскільки саме він запускає процес побудови меню та містить базову логіку для даного процесу.

Також в новому об’єкті було реалізовано окремий метод “_getAmpHtml” який власне відповідає за побудову самого меню сайту. Даний метод надано в додатку Г.

Як видно з коду, в методі досить активно використовується стандартні методи оригінального об’єкту, наприклад для вибору дочірніх категорій меню та отримання поточного рівня меню. Слід зауважити що даний метод проходиться рекурсивно по усіх ввімкнених категоріях продуктів, таким чином в меню може виводитись велика кількість категорій незалежно від їх рівня вкладеності.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 52 |

В загальному використання рекурсивних методів не дуже хороша практика оскільки в разі поганої реалізації такі методи можуть викликати досить великі навантаження на сервер, проте тут без цього не вийшло уникнути. Справа в тому що Magento досить потужна платформа яка може містити дуже велику кількість продуктів, і відповідно велику кількість категорій які в собі можуть містити дочірні категорії. Як варіант можна було б обмежити вивід категорій в меню двома – трьома рівнями, проте це не дуже хороша ідея оскільки в такому випадку користувачі не матимуть доступу до категорій з більшим рівнем вкладеності.

Проте наявність великого списку категорій в меню робить його досить громістким і відповідно не дуже зручним для використання на мобільних пристроях. Щоб уникнути подібних проблем було вирішено для виводу дочірніх категорій використати спеціальний тег “amp-accordion”. Даний тег дозволяє формувати список який буде можливо в подальшому буде можна згорнути та розгорнути безпосередньо вже на самій сторінці. Таким чином це дасть змогу стандартно приховати усі дочірні категорії і відповідно зменшити висоту меню сайту.

Власне за найбільш базові роботи по перебору і оптимізації контенту сторінок відповідає наведена вище реалізація, але на цьому процес оптимізації не закінчується. В реалізованих об'єктах активно використовувалися спеціалізовані AMP теги, які за звичайних умов не будуть працювати і через них виникнуть помилки. В даний момент мова йдеться про теги типу “amp-accordion”, “amp-sidebar”, “amp-list”, “amp-carousel” та інші. Проблема в том, що дані теги для своєї коректної роботи вимагають наявності спеціального JavaScript коду, який слід підключати окремо.

В платформі Magento є досить багато способі підключати власний JavaScript код, але більшу частину з них ми усунули в цілях оптимізації. З цієї причини було вирішено реалізувати в розробленому програмному модулі спеціальний об'єкт “Dathard\Amp\Block\Page\Head\Js” який автоматично перевіряє згенерований код та видаляє проблемні ділянки коду:

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 53 |

```

<?php

namespace Dathard\Amp\Block\Page\Head;

class Js extends \Magento\Framework\View\Element\Template
{
    protected $_js = [];

    protected $_html;

    /**
     * @param $src
     * @param $type
     * @return $this
     */
    public function addJs($src, $type) {
        $this->_js[$type] = $src;
        return $this;
    }

    /**
     * @return string
     */
    protected function _toHtml()
    {
        $this->_html = '';
        foreach ($this->_js as $type => $src) {
            $this->_html .= '<script async '. ($type ?
'custom-element="' . $type . '"' : '') . ' src="' . $src .
'"></script>';
        }

        $this->_html .= '<script async
src="https://cdn.ampproject.org/v0.js"></script>';
        return $this->_html;
    }
}

```

Даний метод має лише методи “addJs” та “_toHtml”. Власне метод “addJs” відповідає за накопичення посилань на JavaScript файли які слід підключити. В свою чергу метод “_toHtml” вже спрацьовує при самій генерації сторінки та обробляє накопичену інформацію про JavaScript файли які потрібно підключити. Проте лише реалізації даного об’єкту мало, його слід підключити щоб він спрацьовував при генерації сторінок, для цього було вирішено скористатися макетами сторінок.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 54 |

В платформі Magento генерація сторінок сайту відбувається згідно до макетів даних сторінок. Макети являють собою спеціалізовані XML файли які містять в собі інформацію про структуру сторінок, які об'єкти і де саме на сторінці вони виводяться та за вивід якого контенту вони відповідають. Власне в даних макетах з допомогою спеціальних тегів можливо вказувати який метод запустити і що в нього передати, це і було використано для роботи з об'єктом "Dathard\Amp\Block\Page\Head\Js". Код макету який відповідає за роботу з даним об'єктом надано в додатку Д.

Даних операцій в загальному має вистачити для оптимізації основної частини контенту сторінок, але не усього оскільки усе залежить від типу сторінок. На даний момент було вирішено реалізувати підтримку лише двох типів сторінок: сторінок продуктів та CMS сторінок. Власне з даних типів сторінок CMS сторінки є найбільш простими оскільки містять звичайний HTML код який можна редагувати з адмінки і а даний момент підтримка таких сторінок практично реалізована повністю.

З оптимізацією сторінок продуктів виникають проблеми оскільки в собі вони зазвичай містять слайдери для фото продуктів. Власне дані слайдери працюють виключно за рахунок JavaScript коду, але оскільки його було видалено то вони вже не працюють. Ситуація з слайдерами ідентична до ситуації з навігацією сайту і для її вирішення потрібно реалізовувати окремий слайдер.

Для реалізації слайдерів було вирішено використати спеціальний тег "amp-carousel". Даний тег слугує свого роду заміною подібних слайдерів для фото та іншого контенту, і є значно оптимізованим ніж його аналоги які реалізовані на базі JavaScript бібліотек, які і без того досить важкі. Власне заміну стандартного слайдера було реалізовано видаливши старий шаблон (phtml файл з HTML кодом) з допомогою макетів сторінок даного типу і замінивши його новим. Новий макет для виводу зображень було реалізовано на базі засобів які надаються технологією AMP, а саме з використанням спеціалізованих тегів та стандартних для технології скриптів. Код має наступний вигляд:

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 55 |

```

<?php
$imageWidth = $block-
>getImageAttribute('product_page_image_medium', 'width');
$imageHeight = $block-
>getImageAttribute('product_page_image_medium', 'height');
$images = $block->getGalleryImages();
?>
<div class="amp-product-gallery">
    <?php if (count($images) > 1) : ?>
        <amp-carousel width="<?= $imageWidth; ?>" height="<?=
$imageHeight; ?>" layout="responsive" type="slides" autoplay
delay="5000" >
            <?php foreach($images as $image) : ?>
                <?php if (! $image->getData('disabled')): ?>
                    <amp-img
                        src="<?= $image-
>getData('large_image_url') ?>"
                        width="<?= $imageWidth ?>"
                        height="<?= $imageHeight ?>"
                        layout="responsive"
                    ></amp-img>
                <?php endif; ?>
            <?php endforeach; ?>
        </amp-carousel>
    <?php else : ?>
        <amp-img width="<?= $imageWidth ?>"
            height="<?= $imageHeight ?>"
            layout="responsive"
            src="<?= $block->getImage($this->getProduct(),
'product_page_image_medium')->getImageUrl(); ?>" ></amp-img>
    <?php endif; ?></div>

```

3.3 Тестування та апробація розробленого програмного модуля

В даний момент в модулі оптимізації поки було реалізовано підтримку сторінок двох типів – сторінок продуктів та CMS сторінок. Модуль автоматично перехоплює згенерований платформою Magento HTML та переформатовує його згідно до вимог технологій AMP.

Для порівняльної характеристики різного роду засобів оптимізації платформи Magento з розробленим було використано спеціалізовані сервіси по

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 56 |

замірах швидкості сторінок вайту - Google PageSpeed та web.dev. Власне отримані показники було зведено у таблиці 3.1 та 3.2.

Таблиця 3.1 – Показники швидкості сторінки продуктів.

| Показники | Стандартні налаштування | З мініфікацією JavaScript та CSS | AMP сторінка |
|---------------------------|-------------------------|----------------------------------|--------------|
| Перша змістовна фарба | 5,0 сек. | 4,4 сек. | 1,3 сек. |
| Індекс швидкості | 22,1 сек. | 19,4 сек. | 1.9 сек. |
| Найбільша змістовна фарба | 9,7 сек. | 6,8 сек. | 3.2 сек. |
| Час до інтерактивності | 38,0 сек. | 37,3 сек. | 3,8 сек. |
| Загальний час блокування | 570 мс | 340 мс | 280 мс |
| Сукупний зсув макета | 0,366 | 0,367 | 0 |

Таблиця 3.2 – Показники швидкості CMS сторінок.

| Показники | Звичайна сторінка | З мініфікацією JavaScript та CSS | AMP сторінка |
|---------------------------|-------------------|----------------------------------|--------------|
| Перша змістовна фарба | 4,9 сек. | 4,4 сек. | 1,9 сек. |
| Індекс швидкості | 19,5 сек. | 16,5 сек. | 2,4 сек. |
| Найбільша змістовна фарба | 9,6 сек. | 6,6 сек. | 3,4 сек. |
| Час до інтерактивності | 34,5 сек. | 34,1 сек. | 3,1 сек. |
| Загальний час блокування | 240 мс | 180 мс | 120 мс |
| Сукупний зсув макета | 0 | 0 | 0 |

Для зняття показників швидкості було використано сервіси “PageSpeed Insights”, “web.dev” та “GTMetrix” оскільки вони найбільш популярними сервісами даного типу і надають найбільш коректні показники.

Як видно із наведених в таблицях результатів замірів стандартні налаштування Magento видають не дуже хороші показники по швидкості роботи сайту, особливо коли на платформу додатково встановлено сторонні модулі. Стандартно Magento надає досить хороші можливості кешування

контенту, але це стосується лише виконання php коду, і в крім того в даний процес можуть втручатися сторонні модулі і уповільнювати його. Причина даних показників в тому, що досить багато часу займає завантаження JavaScript та CSS файлів, власне вони уповільнюють швидкість сайту. В даному випадку досить вагоме покращення показників відбувається саме при оптимізації JavaScript та CSS файлів (мініфікації контенту, їх об'єднанні), це чудово видно з показників таблиці.

Крім проблеми з статичним вмістом також в Magento наявні пробелми з досить громістким HTML, який до речі не дуже добре оптимізований. В процесі реалізації модуля оптимізації було частково усунено дану проблему за рахунок заміни громістких конструкцій (меню, слайдери) та заміни значної частини HTML тегів на більш оптимізовані AMP аналоги. В результаті, як видно із замірів, це дало досить вагомий результат в пришвидшенні роботи сайту.

Слід зауважити що дані показники AMP сторінок знімалися безпосередньо з самого сайту, але коли дані сторінки попадуть в Google кеш то ще покращаться в кілька разів.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 58 |

4 ТЕХНІКО-ЕКОНОМІЧНИЙ РОЗДІЛ

У даному розділі кваліфікаційної роботи проводиться економічне обґрунтування доцільності розробки програмного модуля оптимізації сайту для платформи Magento 2. Зокрема, здійснюється розрахунок витрат на розробку даного програмного продукту, експлуатаційних витрат, ціни на споживання проектного рішення, визначаються показники економічної ефективності нового програмного продукту, обґрунтовуються відповідні висновки.

4.1 Розрахунок витрат на розробку програмного забезпечення

Витрати на розробку і впровадження програмних засобів (K) включають [30]:

$$K = K_1 + K_2, \quad (4.1)$$

де K_1 – витрати на розробку програмних засобів, грн.;

K_2 – витрати на відлагодження і дослідну експлуатацію програми рішення задачі на комп'ютері, грн.

Витрати на розробку програмних засобів включають:

- витрати на оплату праці розробників (ВОП);
- витрати на відрахування у спеціальні державні фонди (Вф);
- витрати на покупні вироби (Пв);
- витрати на придбання спецобладнання для проведення експериментальних робіт (Об);
- накладні витрати (Н);
- інші витрати (Ів).

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 59 |

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудомісткості відповідних робіт у людино-днях та середньої ЗП відповідних категорій працівників.

У розробці проектного рішення задіяні наступні спеціалісти-розробники, а саме програміст та тестувальник.

Витрати на оплату праці розробників проекту визначаються за наступною формулою (4.2):

$$B_{ОП} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij}, \quad (4.2)$$

де n_{ij} – чисельність розробників i -ої спеціальності j -го тарифного розряду;
 t_{ij} – затрачений час на розробку проекту співробітником i -ої спеціальності j -го тарифного розряду, год;

C_{ij} – годинна ставка працівника i -ої спеціальності j -го тарифного розряду.

Середньо годинна ставка працівника може бути розрахована за такою формулою (4.3):

$$C_{ij} = \frac{C_{ij}^0 (1 + h)}{PЧ_i}, \quad (4.3)$$

де C_{ij} – основна місячна заробітна плата розробника i -ої спеціальності j -го тарифного розряду, грн.;

h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$PЧ_i$ - місячний фонд робочого часу працівника i -ої спеціальності j -го тарифного розряду, год. (приймаємо 168 год.).

Результати розрахунку зведено у таблиці 4.1.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 60 |

Таблиця 4.1 – Підрахунки щодо заробітної плати працівників.

| № п/п | Посада виконавців | Час розробки, год | Погодинна заробітна плата, грн/год. | Витрати на розробку, грн |
|-------|-------------------|-------------------|-------------------------------------|--------------------------|
| 1 | Програміст | 280 | 83,3 | 23324 |
| 2 | Тестувальник | 100 | 59,5 | 5950 |
| Разом | | | | 29274 |

Відрахування на соціальні заходи. Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 18% від суми заробітної плати:

$$B_{\phi} = \frac{18}{100} \cdot 29274 = 5269,32 \text{ грн.}$$

Усі проведені обрахунки щодо заробітної плати працівників зведено у таблицю 4.2.

Таблиця 4.2 – Загальні підрахунки щодо заробітної плати працівників.

| № п/п | Посада виконавця | Оклад, грн/міс | Відрахування грн | Кількість | | Сума з/п, грн. |
|------------------|------------------|----------------|------------------|-----------|-----|----------------|
| | | | | чол. | год | |
| 1 | Програміст | 14000 | 4198.32 | 1 | 280 | 23324 |
| 2 | Тестувальник | 10000 | 1071 | 1 | 100 | 5950 |
| Усього зарплати: | | | | | | 29274 |

Розрахунок витрат на матеріали та комплектуючі.

Загальна сума витрат на матеріальні ресурси (B_M) визначається за формулою (4.3):

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 61 |

$$B_M = \sum_{i=1}^n K_i \cdot C_i, \quad (4.3)$$

де K_i – витрата i -го типу матеріалу, натуральні одиниці вимірювання;

C_i – ціна за одиницю i -го типу матеріалу, грн.;

i – тип матеріального ресурсу;

n – кількість типів матеріальних ресурсів.

Таблиця 4.3 – Зведені розрахунки матеріальних витрат

| № п/п | Найменування матеріальних ресурсів | Факт. витрачено матеріалів | Ціна за одиницю, грн. | Сума, грн | Транспортні витрати (10% від суми) | Загальна сума, грн |
|-------|------------------------------------|----------------------------|-----------------------|-----------|------------------------------------|--------------------|
| 1. | Допоміжна література | 1 | 500 | 500 | 50 | 550 |
| 2. | A4 | 80 | 2,5 | 300 | 30 | 330 |
| 3. | Ручка кулькова | 2 | 10 | 20 | 2 | 22 |
| 4. | Олівець простий | 2 | 10 | 20 | 2 | 22 |
| 5. | Тонер для принтера | 1 | 80 | 80 | 8 | 88 |
| 6. | Зошит, 48 арк. | 2 | 25 | 50 | 5 | 55 |
| Разом | | | | | | 1067 |

Витрати на використання комп'ютерної техніки.

До витрат на використання комп'ютерної техніки включають:

- витрати на амортизацію комп'ютерної техніки;
- витрати на користування програмним забезпеченням;
- витрати на електроенергію що споживається комп'ютером.

За статистикою вартість роботи середньостатистичного ПК на сьогоднішній день становить 15 гривень за годину.

Розрахунок усіх витрат на використання комп'ютерної техніки приведений в таблиці 4.4.

Таблиця 4.4 – Розрахунок витрат на використання комп'ютерної техніки

| № п/п | Назва етапів робіт, при виконанні яких використовується комп'ютер | Час використання комп'ютера, год. | Витрати на використання комп'ютера грн. |
|-------|---|-----------------------------------|---|
| 1 | Проведення досліджень та реалізація макету проекту | 90 | 1350 |
| 2 | Реалізація розробленого макету проекту на платформу Magento | 190 | 2850 |
| 3 | Тестування проекту | 100 | 1500 |
| Разом | | 380 | 5700 |

Накладні витрати.

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати. Вони розраховуються за встановленими відсотками до витрат на оплату праці. Середньостатистичний відсоток накладних витрат приймемо 100% від заробітної плати:

$$H = 1 \cdot 29274 = 29274 \text{ (грн.)}$$

Інші витрати.

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I_B = 0,1 \cdot 59040 = 2927,4 \text{ (грн.)}$$

Витрати на розробку програмного забезпечення складають:

$$K_1 = B_{ОП} + B_{\Phi} + B_M + H + I_B + B_{КТ},$$

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 63 |

$$K_1 = 29274 + 5269,32 + 1067 + 29274 + 2927,4 + 5700 = 73511.72 \text{ (грн)} .$$

4.2 Розрахунок експлуатаційних витрат і ціни споживання

Витрати на відлагодження і дослідну експлуатацію програмного продукту визначаємо за формулою (4.4):

$$K_2 = S_{м.г.} \cdot t_{від} \quad (4.4)$$

де $S_{м.г.}$ – вартість однієї машино-години роботи ПК, грн./год;

$t_{від}$ – комп'ютерний час, витрачений на відлагодження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 30 днів. Середній щоденний час роботи на комп'ютері – 2 години. Вартість години роботи комп'ютера дорівнює 15 грн., тому $K_2 = 15 \cdot 60 = 900$ грн.

Обчислення накладних витрат.

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 60–100 % від суми основної та додаткової заробітної плати працівників.

$$H_B = 0,7 * B_{ОП} = 0,7 * (c_1 * 168), \quad (4.7)$$

де H_B – накладні витрати.

$$H_B = 0,7 * 29274 = 20491,8 \text{ грн.}$$

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 64 |

Складання кошторису витрат та визначення собівартості. Результати проведених розрахунків зведемо у таблицю 4.5.

Таблиця 4.5 – Кошторис витрат ($B_{КС}$)

| № п/п | Найменування витрат | Сума витрат, грн. |
|-------|---|-------------------|
| 1 | Витрати на оплату праці ($B_{ОП}$) | 29274 |
| 2 | Відрахування у спеціальні державні фонди ($B_{Ф}$) | 5269,32 |
| 3 | Витрати на матеріали та комплектуючі ($B_{М}$) | 1067 |
| 4 | Накладні витрати на розробку (H) | 29274 |
| 5 | Інші витрати (I_B) | 2927,4 |
| 6 | Витрати на відлагодження і дослідну експлуатацію програмного продукту (K_2) | 900 |
| 7 | Накладні витрати експлуатацію (H_B) | 20491,8 |
| Разом | | 103840,52 |

Розрахунок ціни проекту.

Договірна ціна ($Ц_D$) для проектних рішень розраховується за формулою (4.8):

$$Ц_D = B_{КС} \cdot \left(1 + \frac{p}{100}\right), \quad (4.8)$$

де $B_{КС}$ – кошторисна вартість, грн.;

p – середній рівень рентабельності, % (приймаємо 20% за погодженням з керівником).

$$Ц_D = 103840,52 \cdot (1 + 0,2) = 124608,6 \text{ грн.}$$

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 65 |

4.3 Визначення показників економічної ефективності

Економічна ефективність (E_{ϕ}) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_{\phi} = \frac{\Pi}{B_{КС}} , \quad (4.9)$$

де $\Pi = Ц_{Д} - B_{КС}$ – прибуток, грн.;

$B_{КС}$ – кошторисна вартість, грн..

$$E_{\phi} = 20768.08 \text{ грн.} / 103840,52 \text{ грн.} = 0,2.$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (T_P):

$$T_P = \frac{1}{E_P} . \quad (4.10)$$

Тобто: $T_P = 1/0,2 = 5\text{р.}$

Прийнятним вважається термін окупності близький до 7 років.

Розраховані економічні показники проекту занесемо до таблиці 4.6.

Таблиця 4.6 – Економічні показники розробки

| № п/п | Показник | Значення |
|-------|-------------------------|-----------|
| 1. | Собівартість, грн. | 103840,52 |
| 2. | Плановий прибуток, грн. | 20768.08 |

Продовження таблиці 4.6

| | | |
|----|-------------------------|----------|
| 3. | Ціна, грн. | 124608,6 |
| 4. | Економічна ефективність | 0,2 |
| 5. | Термін окупності, рік | 5 |

Враховуючи основні економічні показники з таблиці 4.6, можна зробити висновок, що при терміні окупності проекту в 5 років. Крім того ціна впровадження проекту значно нижча в порівнянні з іншими технологіями при досягненні ідентичних результатів, що свідчить про доцільність розробки та впровадження розробленого проекту.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 67 |

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було розглянуто наскільки швидкість завантаження web-сайту впливає на його позиції в пошуковій видачі, коефіцієнт конверсії та інші важливі для бізнесу аспекти. Власне даний показник є одним з найбільш вагомих факторів від яких залежить успішність web-сайту.

Також в кваліфікаційній роботі також було детально проаналізовано фактори від яких залежить швидкість завантаження сторінок web-сайту. До даних факторів в першу чергу відносяться:

- вага контенту сторінки;
- відстань між користувачем та сервером web-сайту;
- швидкість виконання серверних скриптів.

В кваліфікаційній роботі було проведено аналіз наявних на даний момент рішень з оптимізації даних факторів та проведено їх проівняння з технологією AMP і дана технологія дозволила отримати найбільш вражаючий приріст в швидкості роботи.

В рамках кваліфікаційної роботи було розроблено програмний модуль який дозволяє адаптовувати згенерований HTML контент платформою Magento згідно до вимог технології AMP. Реалізований програмний модуль автоматично здійснює конвертацію контенту без потреби втручання спеціалізованого персоналу. Керування програмним модулем виконується з панелі керування платформи Magento.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 68 |

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ашманов І. С. Оптимізація і просування в пошукових системах. / Ашманов І. С., 2016. 512 с.
2. Мережа підприємства з безпроводним доступом. Науково–практична конференція молодих вчених і студентів “Інтелектуальні комп’ютерні системи та мережі” / Шевчук Р.В., Полотнянко Р.Я., 2021. 9 с.
3. Побудуйте професійний сайт самі / Борисов В. Д., 2009. 304 с.
4. Побудуйте професійний сайт самі / Байков В. Д., 2012. 288 с.
5. Веб-дизайн і комп’ютерна графіка. Навчальний посібник / Романюк О.Н., Кательніков Д.І., Косовець О.П. , 2007. 147 с.
6. Створення інтерактивного Web-сайту. Навчальний курс. / Лещев Д., 2003. 544 с.
7. РНР 5. Практика створення Web-сайтів. / Кузнєцов М., Симдянов І., Голишев С., 2007. 960 с.
8. Основи програмування на РНР / Ульман Л., 2001. 288 с.
9. РНР 5: повне керівництво / Коггзолл Д., 2009. 752 с.
10. РНР: навчання на прикладах / Кухарчик А., 2004. 237 с.
11. РНР: настільна книга програміста / Мазуркевич А.М., 2003. 480 с.
12. Пошукова оптимізація. Практичний посібник з просування сайту в Інтернеті / Севостьянов І.О., 2016. 272 с.
13. Чек-лист технічної оптимізації сайтів. SEO-книга з просування / Житников Т., 2021. 29 с.
14. Розробка веб-сайтів для мобільних пристроїв / Джейсон Грігсби, Ліза Гарднер, 2013. 448 с.
15. Magento PHP Developer's Guide / Аллан Макгрегор, 2013. 256 с.
16. Magento 2 Cookbook / Володимир Керкхофф, Рей Богман, 2016. 342 с.
17. Magento Search Engine Optimization / Роберт Кент, 2014. 132 с.
18. Magento Mobile How-To / Дарко Голес, 2012. 78 с.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 69 |

19. Magento 2 Theme Design / Fernando J. Miguel, Річард Картер, 2016. 288 с.
20. Magento 2 Development Quick Start Guide / Branko Ajzele, 2018. 218 с.
21. Mastering Magento 2 - Second Edition / Брет Уильямс, Джонатан Баундс, 2016. 340 с.
22. Magento 2 Development Essentials / Fernando J. Miguel, 2016. 178 с.
23. Learning Magento 2 Administration / Брет Уильямс, Синди Уильямс, 2016. 300 с.
24. The Definitive Guide to Magento / Адам Маккомбс, Роберт Бан, 2010. 328 с.
25. Magento Beginner's Guide / Роберт Равенсберген, Сандер Шоневиль, 2013. 320 с.
26. Mastering Magento Theme Design / Андреа Сакка, 2014. 310 с.
27. Grokking Magento: Basics & Request Flow / Винай Копп, 2014. 170 с.
28. Instant E-Commerce with Magento: Build a Shop / Branko Ajzele, 2013. 52 с.
29. The Definitive Guide to Magento / Адам МакКомбс, Роберт Бан, 2009. 328 с.
30. Pro PHP Security : From Application Security Principles to the Implementation of XSS Defenses / Кріс Снайдер, Томас Майер, Майкл Саутвелл, 2011. 368 с.
31. PHP Cookbook / Дэвид Склар, 2014. 80 с.
32. Essential PHP Security / Кріс Шифлетт, 2005. 124 с.
33. Learning PHP, MySQL & JavaScript 5e : With jQuery, CSS & HTML5 / Робин Никсон, 2018. 800 с.
34. Modern PHP: New Features and Good Practices / Джош Локхарт, 2015. 270 с.
35. PHP: объекты, шаблоны и методики программирования. / Метт Зандстра, 2019. 736 с.
36. Чистый код / Роберт Мартін, 2019. 416 с.
37. Выразительный JavaScript / Марин Хавербеке, 2020. 472 с.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 70 |

38. Getting Started with Varnish Cache / Тийс Ферин, 2017. 150 с.
39. Redis 4.x Cookbook : Over 80 hand-picked recipes for effective Redis development and administration / Пэнчен Хуанг, Цзофей Ван, 2018. 382 с.
40. Redis Cookbook : Practical Techniques for Fast Data Manipulation / Тьяго Маседо, Фред Оливерия, 2011. 76 с.
41. Designing Data-Intensive Applications : The Big Ideas Behind Reliable, Scalable, and Maintainable Systems / Мартин Клеппманн, 2017. 562 с.
42. Redis Essentials / Максвелл Дайвсон ДаСильва, Хьюго Лопес Таварес, 2015. 230 с.
43. Docker for Developers : Develop and run your application with Docker containers using DevOps tools for continuous delivery / Ричард Буллингтон-Макгуайр, Эндрю К. Деннис, Майкл Шварц, 2020. 468 с.
44. Google AMP - Accelerated Mobile Pages : A Hands-on, Example-Rich Introduction to Google AMP (Accelerated Mobile Pages) for Beginners / Google Publishing, 2019. 588 с.
45. AMP: Building Accelerated Mobile Pages Руадан Донохью, 2017. 370 с.
46. E-Commerce Website Optimization / Дэн Кроксен-Джон, Юганн Ван Тондер, 2020. 272 с.
47. Bandit Algorithms for Website Optimization / Джон Майлс Уайт, 2013. 80 с.
48. 50 ways to confidently improve your business website / Стюарт Крейг Ловатт, 2013. 50 с.
49. Magento Site Performance Optimization / Матье Найроллес 2014. 92 с.
50. Death of A Website / Дуайния Грей, 2020. 148 с.
51. Мельничук Д.С. Оптимізація швидкості сторінок вебсайтів на базі технології AMP. Науково–практична конференція молодих вчених і студентів «Інтелектуальні комп’ютерні системи та мережі». Тернопіль, 2021. с. 25
52. Методичні рекомендації до виконання кваліфікаційної роботи з освітнього ступеня “Бакалавр” спеціальності 123 «Комп’ютерна інженерія» галузі знань 12 Інформаційні технології / О.М. Березький, Л.О.Дубчак, Г.М. Мельник, Ю.М. Батько / Під ред. О.М. Березького. Тернопіль: ЗУНУ, 2020. 60с.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 71 |

53. Методичні вказівки до виконання практичних робіт з дисципліни «Техніко-економічне обґрунтування розробки комп'ютерних систем»/ Н.Я. Савка, І.Р. Паздрій / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 40 с.

54. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія» / І.В. Гураль, Л.О. Дубчак / Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 33 с.

| | | | | | | |
|------|------|----------|--------|------|-----------------------------|------|
| | | | | | КР.КІ.07177/17.00.00.000 ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата | | 72 |